PUBLICATIONS OF
THE UNIVERSITY OF EASTERN FINLAND

Dissertations in Forestry and Natural Sciences

HIMAT SHAH

Automatic Keyword Extraction for Webpages

Automatic Keyword Extraction for Webpages

Himat Shah

Automatic Keyword Extraction for Webpages

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 485

University of Eastern Finland
Joensuu
2022

Academic dissertation
To be presented by permission of the Faculty of Science and Forestry for public examination in the Auditorium Futura, F100 in the Futura Building at the University of Eastern Finland, Joensuu on 15 th, November 2022, at 12 o'clock noon

Punamusta Oy
Joensuu, 2022
Editors: Pertti Pasanen, Nina Hakulinen, Raine Kortet,
Matti Tedre, and Jukka Tuomela
Myynti: Itä-Suomen yliopiston kirjasto

ISBN: 978-952-61-4687-4 (Print)
ISBN: 978-952-61-4688-1 (PDF)
ISSNL: 1798-5668
ISSN: 1798-5668
ISSN: 1798-5676 (PDF)

Author's address: Himat Shah
School of Computing
University of Eastern Finland
P.O. Box 111
80101 JOENSUU, FINLAND
email: himat@cs.uef.fi

Supervisors: Professor Pasi Fränti, Ph.D.
School of Computing
University of Eastern Finland
P.O. Box 111
80101 JOENSUU, FINLAND
email: franti@cs.uef.fi

Reviewers: Professor Tapio Salakoski, Ph.D.
Computer Science
University of Turku
Vesilinnantie 5
20500, TURKU, FINLAND
email: tapio.salakoski@utu.fi

Professor Vasile Manta, Ph.D.
Technical University of Lași Faculty of Automatic Control and Computer Engineering
Department. of Computer Engineering Blvd. D. Mangeron 53A, 700050, LAȘI, ROMANIA
email: vmanta@cs.tuiasi.ro

Opponent: Professor Jyrki Nummenmaa
Tampere University Faculty of Information Technology and Communication Sciences
Department of Computing Sciences
city center campus, TAMPERE, FINLAND
email: Jyrki.nummenmaa@tuni.fi

Shah, Himat
Automatic Keyword Extraction for Webpages
Joensuu: University of Eastern Finland, 2022
Publications of the University of Eastern Finland
Dissertation in Forestry and Natural Sciences; 485
ISBN: 978-952-61-4687-4 (Print)
ISSNL: 1798-5668
ISSN: 1798-5668
ISBN: 978-952-61-4688-1 (PDF)
ISSN: 1798-5676 (PDF)

ABSTRACT

The quantity of text documents on the Internet has increased so quickly that manual analysis is no longer feasible. Extracting the key elements from studied documents requires an efficient keyword extraction approach. Recent years have seen extensive research in keyword extraction, with applications in text-mining, information retrieval, and natural language processing. A keyword extraction process is the automated extraction of single or multiple token phrases from a textual document. This process supports all key aspects of its content and provides an automated summary of the document.

Automatic keyword extraction is difficult due to the complexity of natural language and the variety of web content and topics. There is a dire need for automated keyword extraction methods that can extract keywords from multiple languages. We present four novel methods for automatic keyword extraction for webpages, including three language-independent methods that can extract keywords from different languages. We analyze the performance of our methods using hard and soft evaluation metrics. Our results improve the current state-of-the-art methods and provide readily available solutions for automatic keyword extraction for webpages.

Keywords: Data mining, keyword extraction, supervised machine learning, web mining, text analysis

ACKNOWLEDGEMENTS

My thesis summarizes the results of research completed during the years 2018-2022 at the School of Computing of the University of Eastern Finland. In appreciation of my supervisor, Professor Pasi Fränti, who has given me the opportunity to work in his group, I am sincerely grateful. Without his never-ending enthusiasm and strong encouragement, I could not have completed this thesis.

I am grateful for Radu Mariescu-Istodor, Mohammad Rezaei's and Muhammad Usman Khan contributions to this research, as well as my co-supervisors with whom I have thoroughly enjoyed working on this study. It would not have been possible for this study to proceed without their constant discussion and guidance.

My sincere gratitude also goes out to my colleagues with whom I have had the pleasure of working during these last few years, particularly Sami Sieranoja, Lahari, Nancy Fazal, Gulraiz Choudhary, Masoud Fatemi, Chengmin Zhou, Nilima Shah, and Abigail Wiafe.

I want to thank Oili Kohonen for the support and care she has provided.

Throughout these years away from home, my parents, brothers, and sisters, have been in my heart and given me strength. Words can not express how much I appreciate my family and all my friends. I would not be able to experience these moments if their constant love and support were not there to support and encourage me.

I am forever indebted to my lovely wife, Laraib, for her tireless support and love during this endeavor and our child, Jafar, whom I adore.

Joensuu, November 2022
Himat Shah

TABLE OF CONTENTS

LIST OF ORIGINAL PUBLICATIONS

In this thesis, the author reviews his work in the field of automatic keyword extraction for webpages and cites selected publications.

P1 H. Shah, M. U. S. Khan, and P. Fränti, H-rank: a keywords extraction method from webpages using POS tags, In IEEE international conference on industrial informatics (INDIN), Helsinki, IEEE, 2019. http://doi.org/10.1109/INDIN41052.2019.8972331

P2 H. Shah, M. Rezaei, and P. Fränti, DOM-based keyword extraction from webpages, In proceedings of international conference on artificial intelligence, information processing and cloud computing (AIIPCC), Sanya, China, Article No. 62, ACM, 2019. https://doi.org/10.1145/3371425.3371495

P3 H. Shah, R. Mariescu-Istodor, P. Fränti, WebRank: language independent extraction of keywords from webpages, In IEEE international conference on progress in informatics and computing (PIC), IEEE, 2021. http://doi.org/10.1109/PIC53636.2021.9687047

P4 H. Shah, P. Fränti, Combining statistical, structural, and linguistic features for keyword extraction from web pages, Applied Computing and Intelligence, 2(2), 115-132, 2022. http://doi.org/ 10.3934/aci.2022007

The papers in this thesis are referred to as [**P1**]-[**P4**] and are included at the end of the thesis with the permission of their original owners.

AUTHOR'S CONTRIBUTION

P1 This paper mimics the idea from CLRank [49].
P2 The idea came from a paper on title extraction [8].
P3 This paper was refined and implemented with the help of Mariescu-Istodor.
P4 Professor Pasi Fränti originated the idea for the paper.

All papers were written by the first author and refined together with the co-authors. In [**P1**], [**P2**], [**P3**] the ideas originate from the first author, whereas the idea for the paper [**P4**] originated from Prof. Pasi Fränti. The first author performed most of the experiments for all papers. The first author also implemented the methods fully for [**P1**], [**P2**],[**P4**] and partially for [**P3**], while the co-authors contributed to the implementation and experimental phases.

ABBREVATIONS

AKE	Automatic Keyword Extraction
CSS	Cascading Style Sheets
DOM	Document Object Model
HTML	Hyper Text Markup Language
JS	JavaScript
K-NN	K-Nearest Neighbors
NLP	Natural language Processing
POS	Part-of-Speech
SVM	Support Vector Machines
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
URL	Uniform Resource Locator
VIPS	Vision-based Segmentation
XHTML	eXtensible HTML
GPE	Geo-Political Entity
WHO	World Health Organization

1 INTRODUCTION

The internet is growing extremely fast and is generating a substantial amount of data. Today, there are 1.92 billion webpages (see Figure 1) and 6 billion internet users.[1] The internet is available to almost everyone regardless of age, including children. The whole world benefits from it: Education, hospitals, entertainment, sports, news, and e-commerce all rely on the internet to function. According to data creation statistics,[2] 2.5 quintillion bytes of data are created every day on the internet.

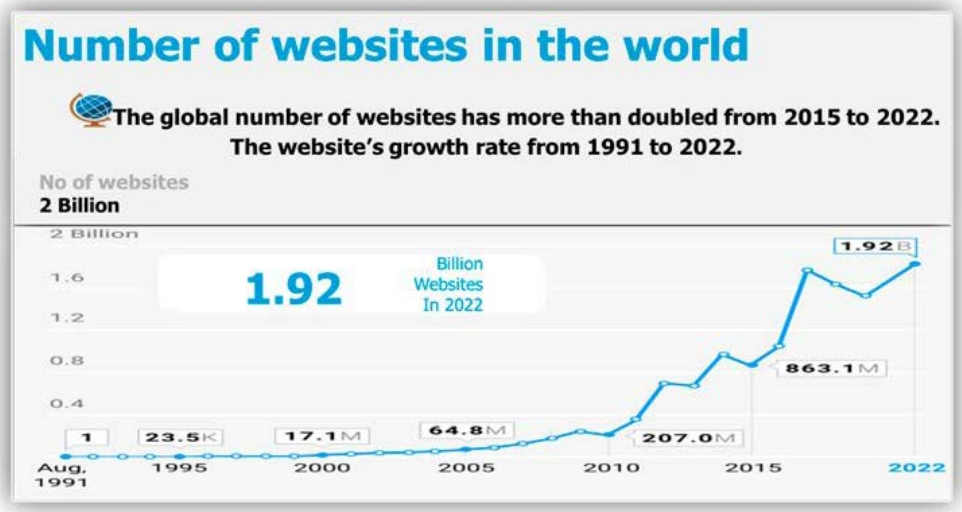

Figure 1. Number of webpages worldwide between 1991 and 2022.

Internet users often rely on Google or one of the many other search engines to reveal useful information for them. Searching for information on search engines is often challenging, time consuming, and confusing [1] considering the volume of data available. Text-mining techniques make it

[1] http://internetlivestats.com
[2] https://earthweb.com

easier for users to find the exact data they need within a short period of time. In fact, it is nearly impossible to locate relevant data without employing text-mining techniques [2] such as automatic keyword extraction, which can help find exact data quickly and accurately.

Google defines a keyword as an isolated word or phrase that provides concise high-level information about content to readers [3]. With the increasing amount of data, users need more resources and time to understand content. Keywords make it easier to understand the meaning of a text in fewer words. In short, keywords summarize the key points presented in the text.

When searching for information on search engines, keywords play a significant role in finding relevant content. Keywords are the most informative part of a text; they are the most prominent words in the text and describe its content [4]. Keywords are necessary in situations involving huge amounts of text data that need to be processed automatically. Keywords are widely used in document summarization, indexing, categorization, and clustering of huge datasets [5]. Many scientific publications contain keyword lists that have been explicitly assigned by their authors. Other documents, however, have not been assigned keywords [6].

As webpages are constantly updated, it is difficult to create keywords manually. Manual keyword assignment is labor intensive, time consuming, and error prone. Specialized curators use fixed taxonomies for manual keyword generation [7], but in some cases, the keywords chosen by the author are not sufficiently comprehensive and accurate. Without high-quality keywords, users fail to catch relevant information [8]. Keywords offer readers a concise high-level summary of a document's content, thereby improving their understanding of that text [9]. Keywords are the most relevant and important indicator for users seeking to grasp the fundamentals of a topic when scanning or skimming an article.

Keyword extraction is a basic step in many text-mining and natural language processing (NLP) techniques, including text summarization, information retrieval, topic modeling [10], clustering [11], and content-based advertisement systems [12]. Finding the relevant webpages, a user is seeking is often a challenging task for which representative keywords or keyphrases

assigned to each webpage are very helpful. Table 1 shows the format of a meta tag that contains keywords.

Table 1. Example of keywords inside a meta tag.

<title>Buy Outdoor Toys, Slides, Ride Ons Trampolines, Indoor Toys, Wooden Toys, Soft </title> <meta name = "description" content = "Buy Outdoor Toys, Slides, Ride Ons Trampolines, Indoor "> < meta name = "keywords" content = "toys, outdoor toys, climbing accessories, climbing frames ">

Keyword extraction in the web documents context entails two main challenges. The first is the presence of noise and irrelevant data, such as navigation bars, menus, comments, and ads (Figure 2). The second is the presence of multiple topics or multiple languages [13]. Besides these, there are other challenges we will discuss later.

Figure 2. Advertisement, and menus are extra information on a webpage [P3].

1.1 The structure of webpage

Webpages have no clear structure and can be presented in a free-form manner, although there is an HTML standard that webpages usually follow. A webpage is generally constructed using three components [16]: HTML, which structures content; CSS, which enables stylization of that content; and JavaScript, which enables interactivity. HTML data is structured like a complex text and data is scattered over different parts. Data on a webpage is considered semi-structured, meaning that they lack an organized structure [15]. HTML tags form a tree structure and can be beneficial in the data extraction process. An internet webpage is an electronic document on the World Wide Web written in HTML and identified by a URL.

The visual structure of today's websites is most often formed by CSS styling. HTML is used to input the webpage content and present it in a semantically correct manner. A webpage's contents are determined by HTML tags, such as heading tags (<H1> through <H6>), image tags (<img>), hyperlink tags <a>, and paragraph tags (<p>), that provide the document with this semantic meaning.

The importance of each item on a webpage can be determined by its tag—for example, the headline on a news article should typically have a <H1> tag, as <H1> is the leading and most important heading [17]. A web browser interprets each tag in sequence to display the page, as shown in Figure 3. Web browsers, for example, assume that all the text after the opening tag will be bold until the closing tag appears. Some tags, such as <title> and <meta>, provide information about the document but do not affect its visual appearance [18, 101].

```
<!DOCTYPEhtml>
<html>
<head>
<meta charset ='UTF-8' />
<title>Universityherald</title>
</head>
<body>
<h1> University Herald news </h1>
<img src = 'herlad.jpg width ='80' hight = '100' />
</body>
</html>
```

Figure 3. HTML structure of a webpage.

CSS defines the format and the layout of a webpage, including how the elements look and appear on the screen [19]. CSS allows documents to be distinguished in terms of content and styling information, which includes components such as layout, colors, and fonts [20]. JavaScript makes webpages interactive by sharing styling files (.css) across multiple webpages.

The Document Object Model (DOM) tree provides fine-grained structural information for a webpage, including content and presentation [21, 69]. The <HTML> tag is the root element of the tree, while the text between <text > and </text> tags encompass the leaf elements. An HTML page can be visualized as a tree formed by parent–child relationships among HTML elements [22, 100]. In the DOM, scripts and programs can access and manipulate the content, structure, and style of a webpage, regardless of their written language. Figure 4 shows an example DOM structure.

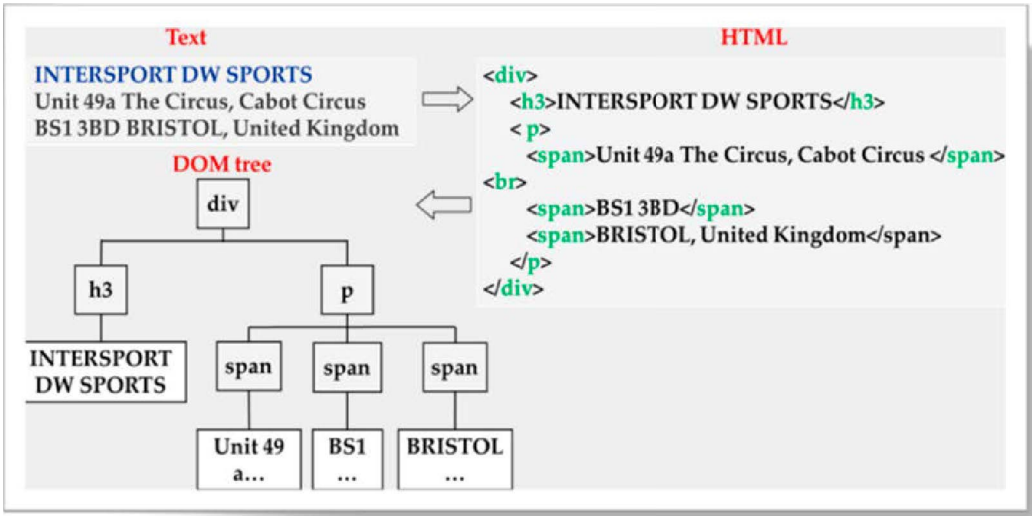

Figure 4. Document Object Model representation of a section on a HTML webpage.

A webpage may include more than one language on the same page. It is difficult to support all domains and languages [23]. It might be difficult to determine which language is the base language, although this information may be necessary during the keyword extraction process. Therefore, we detect the language only after removing stopwords in the preprocessing stage. NLP is time consuming for all languages and is not available at all for some languages [24, 94]. There are many languages in the world, and NLP cannot be limited to only English text.

1.2 Multiple languages and topics on a same page

The second challenge in keyword extraction for webpages is the presence of multiple topics and even multiple languages [26]. Figure 5 shows a university homepage that uses four different languages. Keyword extraction methods find it difficult to deal with these types of webpages. There is a dire need for a general keyword extraction method that works for multiple desired languages.

Figure 5. A Multilingual webpage [P4].

Various keyword extraction methods have been presented in the literature. We divide several of these methods [4, 7, 28, 30, 34, 70, 71] into two categories: (1) keyword extraction from normal text and (2) keyword extraction from a webpage. The normal text is the text that appears in documents other than webpages, such as abstracts, MS Word documents, and so on. Most existing methods [**P1**, 7, 29, 31, 32, 69] from the last few decades are language dependent. Studies on language-independent approaches have been limited because they usually perform worse than methods that take advantage of linguistic features [27, 97]. However, the disadvantage of these methods is that they are only available in a limited number of predefined languages [33]. The language models for all languages may not be freely available, and when they are, they often have distinct representations [35].

The majority of existing keyword extraction methods use language-dependent NLP-based techniques, including part-of-speech (POS) tagging, stemming, and lemmatization, which makes it complex to generalize a method for different languages. Our goal in this research is to extract only language-independent features from webpages, enabling our method to work with different languages. The existing methods of keyword extraction

suffer from high computational complexity or large corpus dependence, which limits their practical use [36, 96].

In keyword extraction, it is challenging to pick the best combination of features from the wide range of options. Which features to choose is an open question for researchers. The addition of more features may increase noise and reduce algorithm performance. Another challenge is to generate a new keyword that does not exist in the text. Generally, a word is selected as a keyword from the text of a webpage only if that word exists in the text.

Keyword and keyphrase words are used interchangeably and have almost the same meaning [37, 93]. Automatic keyphrase extraction is a natural extension of the keyword extraction problem in which phrases are identified as potentially relevant descriptors of a document rather than only identifying unigrams. Keyphrases can be constructed by collapsing co-occurring keywords into phrases as part of postprocessing [31, 95]. Keyword extraction and keyword generation are two subtasks; in keyword extraction, a set of keywords is extracted from the given text, and in keyword generation, a set of keywords is generated from the text as well as from outside the text [38].

In our research, we address keyword extraction difficulties like language independence, webpage structure complexities, and how to select better features. There is an extreme need for a general keyword extraction method that can extract keywords without relying on any specific language. Therefore, we introduce new language-dependent and language-independent keyword extraction methods in our research.

1.3 Research objectives

The research goal is defined as follows. Given a web document as an input, the task is to find a set of words, *keywords*, that best describe the content. Most existing literature focus on extracting keywords from text documents. Less attention has been paid on the challenges involved in web documents. Webpages contain many formatting commands that can disturb the extraction process but also provide additional cues. In this thesis, we apply ranking based framework where candidate keywords are first extracted and then

scored using different features. The final selection of the keywords are the highest scoring candidate words.

2 THE AUTOMATIC KEYWORD EXTRACTION PROCESS

The process of automatic keyword extraction from webpages involves selecting a set of words that best describe the contents of a document. Generally, there are four main steps in the keyword extraction process: (1) preprocessing, (2) candidate generation, (3) feature formation, and (4) ranking and scoring (see Figure 6).

During preprocessing for Chinese and other languages without explicit separators, sentence splitting, tokenization, stemming, and POS tags are applied as a first step. In the second step, keywords are selected and identified to generate a list of candidate keywords. The first two steps can greatly influence the accuracy of the final keyword extraction [36]. We discuss each step-in detail below.

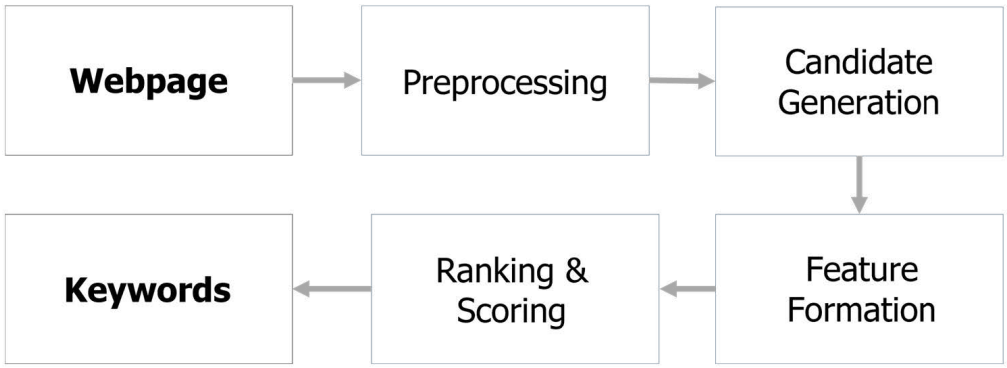

Figure 6. Automatic keyword extraction process.

2.1 Preprocessing

Initially, we downloaded all the content of a webpage, accessed through the DOM structure of the webpage. JavaScript and CSS content was removed because this content is mainly used for webpage formatting.

Preprocessing [**P1–P4**] is the primary step after extracting the content of a webpage. Usually, preprocessing step uses NLP-based techniques such as cleaning, tokenization, stopwords removal, stemming, and lemmatization. A webpage usually contains a range of unrelated content such as navigation menus, decoration, interaction, and contact information. Unstructured webpage data may include numbers, symbols, punctuation marks, hyperlinks, CSS, and JavaScript content [40]. The preprocessing step helps to shape the data properly [39]. The preprocessing of text significantly affects the keywords because the output of this step is the input for the keyword candidate generation phase.

Cleaning is the first step in preprocessing. During cleaning, punctuation marks, symbols, numbers, and special characters are removed. Figure 7 shows the removal of tag symbols (>, <) and punctuation marks (periods) from an example sentence. Regular expressions are often applied to clean noisy data, as regular expressions are easier to implement and faster then other available techniques [41].

Tokenization, which follows the cleaning step, is a process of breaking down text into small lexical units called tokens [42]. During tokenization, white space, line breaks, and punctuation characters are used as separators in order to create individual meaningful entities—usually words—that can then be processed. For keyword extraction, it is important to convert whole text or sentences into smaller units or words before applying other preprocessing steps.

In the next step, we removed stopwords from our list of tokens. Stopwords are frequent or common words in the text [43, 90]. Stopwords strongly depend upon the language and differ across languages. Some examples of stopwords in the English language include “the,” ”and,” “is,” “am,” and “are.” In Figure 7, stopwords such as “back” and “to” are removed after tokenization. We exclude stopwords from text despite their higher frequency because they

are generally not important to the text and do not have any meaning on their own [44].

Usually, lists of stopwords are available in programming libraries, but there are other ways to ascertain stopwords in the text. First, text can be converted into tokens that are then sorted based on number of repetitions in the text. Tokens with the highest frequency can be counted as stopwords. However, it can be difficult to decide how many high-frequency tokens are stopwords, creating the possibility of mistaking important words as stopwords (or vice versa). In our method Hrank [**P1**], we used stopword lists in English and language detection process in others [**P2, P3, P4**] to identify the language and obtain stopwords lists. Removing stopwords can sometimes cause problems, as taking stopwords out of a phrase may cause it to lose its meaning [42].

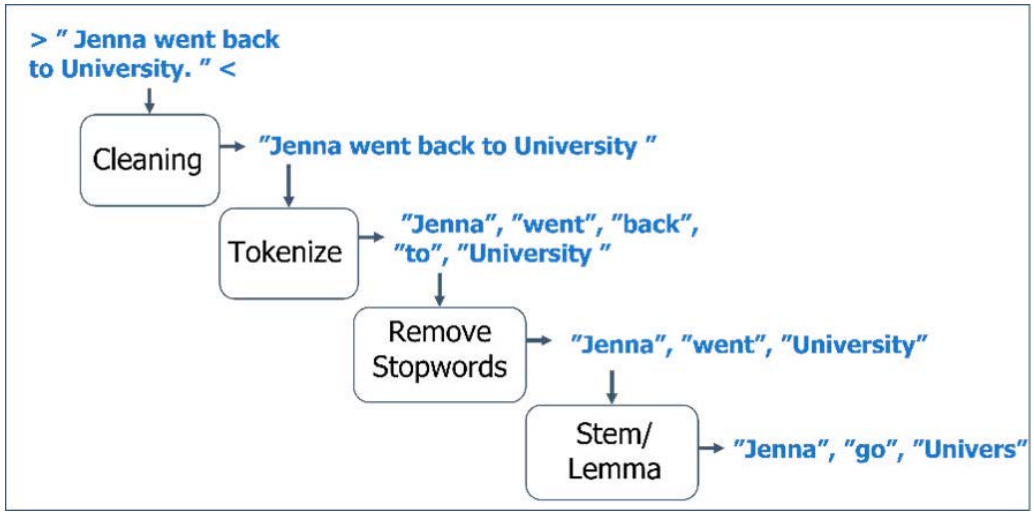

Figure 7. Cleaning, tokenization, stopword removal, and stemming/lemmatization of an example sentence.

In the final step of preprocessing, we normalized the words. Normalization can be achieved using stemming or lemmatization [45]. Stemming refers to the process of reducing inflected words to their stem, base, or root in information retrieval and linguistic morphology [46]. Different languages require different

stemmers due to their dissimilar linguistic structures. The k-core retention algorithm [47] uses stemming and claims to improve performance.

The selected stem is not necessarily the same as the morphological root of a word, and the related words should only be mapped to the selected stem, even though this selected stem is not the main root itself. A stemmer is a software that attempts to perform the stemming process on a selected text. Both stemming and lemmatization can make some sense in the base form of the word. The stemming base form may or may not make sense. Both stemming and lemmatization depend on language. Stemming is widely used, and various stemming algorithms are available: Porter stemmer, Snowball stemmer, and Lancaster stemmer. We can see in the Table 2; word computer is stemmed to comput which is not a proper word. But lemmatization perform well on all the words.

Lemmatization is not as common as stemming because it requires building a dictionary and special language characteristics to build the lemma. Lemmatization chops the last character after identifying it in a dictionary, while stemming does not. Stemming is easier to implement and faster than lemmatization [48].

Table 2. Stemming and lemmatization examples.

Word	Stemming	Lemmatization
information	inform	information
informative	inform	informative
computers	comput	computer
feet	feet	foot

2.2 Candidate generation

In keyword extraction, the candidate generation process follows preprocessing. In this step, specific words from the preprocessed text are chosen to be part of the final keywords. Different combinations of POS tagging have been tested in previous research—for example, nouns + adjectives + nouns, nouns + nouns, and nouns + adjectives + verbs. The method used in [49]

filters nouns using a POS tagger after tokenization. Candidate keywords are selected according to heuristic rules among all possible phrases in the text [50]. Figure 8 show how candidate keywords are separated from the text of the webpage.

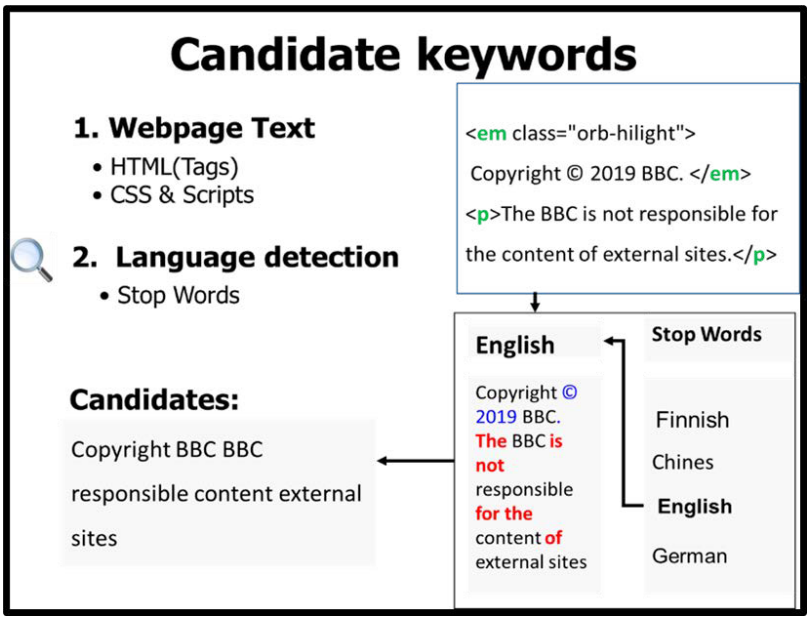

Figure 8. Candidate generation step.

We chose nouns, adjectives, and verbs in [**P1**] and chose all the words or tokens of the text in [**P2**] and [**P3**]. In [**P4**], we tested different combinations of POS tags applied in English datasets. In one method, nouns are used only, while in another they are combined with adjectives and verbs. We observed in [**P4**] that the base method improved after adding linguistic knowledge as well as after adding adjectives and verbs with nouns. Additionally, we used all the words of the text as candidate keywords in the Finnish and German datasets.

2.3 Feature formation

After the candidate generation step, we create a list of features. Each candidate word is checked and assigned a score if it appears in the feature list. Generally, there are three categories of features in automatic keyword extraction: statistical, structural, and linguistic. The choice of features from these categories makes the keyword extraction method either language dependent or language independent [51]. Statistical and structural features are general features that can be used by both language-dependent and language-independent methods. Linguistic features are specific to a given language and cause keyword extraction to be language dependent. Statistical features are widely used and are domain and language independent [52]. Statistical features are more common and are used by many keyword extraction methods. Language models may not be freely available for all languages, and when they are, they often have different representations. Language-independent methods are much easier and faster to implement in practice because they do not require complex NLP components.

There is a direct correlation between the number of features and the performance and efficiency of an algorithm. Simple features are more efficient but may suffer from data sparsity. Inclusion of more features may also create noise and degrade the performance and efficiency of an algorithm [53]. We discuss each of the feature categories later in separate chapters.

2.4 Ranking and scoring

After feature formation, each candidate word is assigned a score. Candidate keywords are ranked, and the top n keywords are selected. In [**P1**], we used hierarchical clustering to extract keywords from a single webpage. After the preprocessing step, nouns, adjectives, and verbs are clustered. We clustered the candidate words based on their semantic relatedness score using WordNet [73, 92]. We used hierarchical clustering because it is easier to control the number of clusters using a simple threshold value.

In [**P2**], we assigned a manual or hard-coded score to the features. Words were scored according to their positions on a webpage—in the URL, title, six levels of headings, and hyperlinks—which provide important information for keyword extraction. We also considered the frequency of words when scoring. Table 3 illustrates features and their manual scores.

Table 3. Manual score for features.

Position of word	**Score**
H1	6
H2	5
H3	4
H4	2
H5	2
H6	2
Title	5
URL Host	5
URL Query	4
Hyperlink (anchor)	2

The final score of a candidate word was calculated as follow:

$$Score = 6 \times 0.2 + 5 + 5 = 11.2$$

Final score is calculated by combining its scores from different location. For example, if a word occurs six times on a webpage and appears in the H2 tag and the title, the score is calculated as above. If the length of words in the text is less than 100, then it will be multiplied by (0.2), otherwise by (0.5).

3 STATISTICAL APPROACH

In keyword extraction, more frequently occurring words are given more weight than those that appear less frequently. The most frequent terms are always important in keyword extraction methods. Statistical measures such as word frequency, term frequency–inverse document frequency (TF–IDF), and n-grams can identify words that repeat and low-importance words that appear evenly in the corpus [54].

Low-frequency words are either removed from the ranking stages or cannot be considered candidate keywords. Nevertheless, if a low-frequency word appears in an important position such as the header, title, or URL of a webpage, it becomes more significant. Statistical methods have the disadvantage that in some professional texts, such as medical and other health-related documents, the most important keyword may only be used once in the article [47, 98].

These methods do not require any training and are domain independent [55]. A combination of linguistic knowledge and statistical features can improve keyword extraction methods. Statistical features were introduced in 1970 and are commonly used by keyword extraction methods. Nowadays, statistical features are considered old features. Term frequency is the most common feature used in keyword extraction [56]. It is defined as the number of times a word or term appears on a webpage. The more times a word or phrase occurs, the better its chances of being considered a keyword. Term frequency is calculated using formula (1):

$$TF = (n \times ti)/N \quad (1)$$

where n × ti is the number of times a term i appears on a webpage and N is the total number of words on the webpage. Term frequencies perform better following preprocessing steps like stopword removal and tokenization.

3.1 Term frequency

Term frequency performs better on webpages due to their heterogeneous nature and the fact that important words are repeated. Moreover, humans tend to choose words that often appear on a page. It fails, however, when some webpages use synonyms to emphasize meaning. In processing long texts, term frequency is both more reliable and more useful than short text. A limitation of this feature is that it does not distinguish between common or grammatical words and content [57].

3.2 Inverse document frequency

IDF is used to indicate how many times a given word has been repeated in other documents when words (e.g., verbs) are repeated many times in several documents but are not keywords. It is calculated using formula (2) given below:

$$IDF_{wi} = \log(\frac{1}{DF_{wi}}) = \log(\frac{Number\ of\ documents\ in\ the\ collection}{Number\ of\ documents\ in\ which\ wi\ occurs}) \qquad (2)$$

where DF is the document frequency of a term corresponding to the number of documents in the collection in which it appears. In general, the IDF is lower for terms that frequently appear in documents in the collection and higher for terms with low DFs. IDF provides a valuable indication of the specificity of a term referring to a document. However, the document needs to be confronted with a collection to use in the IDF. Paper [58] calculates IDF based on tweet frequency and [59] uses inverse webpage frequency.

3.3 TF–IDF

The TF–IDF feature combines term frequency and document frequency and is calculated as in equation (3). TF–IDF features high-score words, which are important words in the text and likely to become keywords.

$$TF - IDF = TF_{wi} \times IDF_{wi} \quad (3)$$

This feature is very common [50, 59–61] in keyword extraction. Table 4 shows an example of calculating TF–IDF for the following phrases:

A: "a new car, used car, car review"
B: "a friend in need is a friend indeed"

Table 4. TF–IDF score calculation for example phrases.

Word	**TF**		**IDF**	**TF–IDF**	
	A	**B**		**A**	**B**
a	1/7	2/8	log (2/2) = 0.0	0.00	0.000
new	1/7	00	log (2/1) = 0.3	0.04	0.000
car	3/7	00	log (2/1) = 0.3	0.13	0.000
used	1/7	00	log (2/1) = 0.3	0.04	0.000
review	1/7	00	log (2/1) = 0.3	0.04	0.000
friend	00	2/8	log (2/1) = 0.3	0.00	0.008
in	00	1/8	log (2/1) = 0.3	0.00	0.040
need	00	1/8	log (2/1) = 0.3	0.00	0.040
is	00	1/8	log (2/1) = 0.3	0.00	0.040
indeed	00	1/8	log (2/1) = 0.3	0.00	0.040

3.4 Word co-occurrence

The co-occurrence feature [31, 62, 63, 64] of a word is a statistical measure that captures words that often occur together. The statistical method used in [65] claims better results than a graph-based method using preprocessed text with statistical features—essentially TF–IDF and word co-occurrence.

4 LINGUISTIC APPROACH

Keyword extraction methods based on a linguistic approach are domain and language dependent [7, 99]. In a linguistic-based approach [12, 50, 55, 82], keywords are extracted using POS tagging and lexical, grammar, syntactic, and term relationship analysis. Among these approaches, the main approaches used by linguistic methods are syntactic (syntax) and semantic (meaning) analysis. Syntactic analysis deals with the grammatical structure of a text. The semantic nature of a text is defined by its intended meaning. For example, “call me Cab” and “my name is Cab” may have syntactical correctness but not semantic correctness.

Linguistic features that aid in understanding terms include POS tagging, chunking, and named entity recognition, which can be added to NLP systems to enhance automatic understanding. In [50, 55] adding linguistic features improved statistical methods significantly. Linguistic keyword extraction methods make use of the POS tagger to detect nouns, adjectives, and verbs within texts as candidate keywords.

In [12], Hulth incorporated linguistic knowledge into the extraction process from noun-phrase chunks (as opposed to frequency and n-grams) and added the POS tags associated with the terms as extra features. It is insufficient to rely only on statistical information to extract important keywords, as this lacks the ability to identify related semantic words acting as synonyms.

4.1 WordNet

WordNet is an English-language corpus that contains words and their relations, expressed in the form of sentences that are essentially synonyms. First, we parsed the entire text’s words, and using WordNet, we obtained relevant words that may be repeated in the text or represented as synonyms. It helps to loop all of them together and then.

4.2 Synonyms and lack of synonyms

Different linguistic methods [11, 26, 49, 61] replace each word with its synonym—for example, "internet" with "net" and "see" with "watch" or "look." In [11], Chinese synonyms were used. In [**P1**] and [**P4**], we used WordNet to find synonyms for candidate words. The method used in [**P4**] identifies a list of words that are missing a synonym in WordNet and considers them as an important feature.

4.3 Part-of-speech tagging

POS tagging determines whether a term is a noun, adjective, verb, or another POS, which are language specific, and is applied to the string parts of a text. Many methods [**P1**,12, 49, 61, 66] employ different combinations of these parts, known as patterns. In a study of the annotations on SemEval [25], 93% of the keywords assigned by readers and professional indexers are nouns. There are many sequences of POS tags used by Hulth [12], including:

- Adjective + noun (singular)
- Noun + noun (singular)
- Adjective + noun (plural)
- Noun (singular) + noun (plural)
- Adjective + noun (plural)

4.4 Wikipedia

In keyword extraction, Wikipedia [50, 60, 67, 68] is used in different ways. In [67], Wikipedia was used to select keywords that were not found on the short-text webpage. If there were not enough keywords on the short-text page, then Wikipedia was used to select keywords. The method used in [68] built a domain-specific concept hierarchy based on Wikipedia, and associated keywords were matched with those concepts.

Using Wikipedia as a corpus of the source text (WIKG), [67] proposed a novel method for flexible keyword generation that took advantage of Wikipedia's rich link structure to construct a graph of entries iteratively, starting with seed keywords. There are many internal links in Wikipedia, and interconnections among concepts and keywords help determine the relevance of articles. Topics are formed according to the relationship between concepts and keywords.

4.5 Named entity

A named entity is one whose POS tag is a proper noun. Named entities can be used to find names of people, places, and organizations across many languages [1]. An example of each type of named entity is shown in Table 5 below.

Table 5. Named entity types with examples.

Named Entity Type	Example
Organization	WHO
Person	President Obama
Location	Mount Everest
Date	10-9-2022
GPE	Northwest America

A named entity scan of an entire article can reveal the major people, organizations, and places discussed in that article. As shown in Table 6, Apple is a proper noun and is an organization, U.K. is a governmental entity, and money is the last word.

Table 6. Example for named entities.

Apple	is	looking	at	buying	U.K.	startup	for	$1 billion
ORG	-	-	-	-	**GPE**	-	-	**MONEY**

Named entities such as persons and places are often used as keywords in web documents. Named entities were tested as a feature in our method [**P4**] but did not provide any significant improvement.

4.6 Semantic similarity

Semantic similarity is a measure of how semantically similar two terms are [50]. Semantic similarity builds on the concept of synonyms. Semantically similar words and phrases do not need to have the exact same meaning, but if the document contains many words with likewise meanings. For example, it supports the similarity between noun pairs (e.g., “cat” and “dog”) and verb pairs (e.g., “run” and “walk”) as well as sentences with similar meanings (e.g., “How old are you?” and “What is your age?”). Figure 9 shows words and their semantic relations to a family of words.

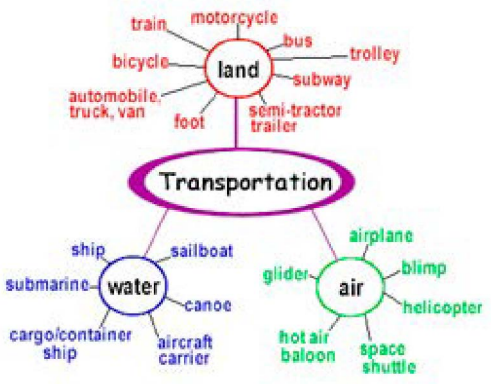

Figure 9. Families for words using semantic similarity.

There is a difference between semantic similarity and semantic relatedness. Unlike semantic similarity, the similarity is not based on concepts such as antonymy and meronymy. Using semantic similarity [84] for evaluation is difficult because perfect semantic senses are only understood under specific circumstances. A semantic similarity score was calculated between two words using WordNet in [P1, 49, 91]. WordNet relations are non-hierarchical.

4.7 Co-occurrence window

A co-occurrence of a word describes when two words that occur frequently together create a keyphrase. A co-occurrence relationship is controlled by the distance between word occurrences: Two vertices are connected if their corresponding lexical units co-occur within a window of time [36]. For example, ”World Cup” refers to an international sports event; if it is separated into the words “World” and “Cup,” then its semantic meaning would be changed.

4.8 Transformation

The keyword of an article rarely changes and usually remains constant throughout the entire article. However, for insignificant words, synonyms are used to prevent uniformity in the article. To calculate this feature in [1], all synonyms of a word in Persian were first derived using FarsNet.

5 STRUCTURAL APPROACH

The DOM enables the division of webpages into meaningful segments, calculation of language-independent feature vectors for each word, and production of a classifier model that identifies how likely each word is to appear. Based on the assumption that the most important information occurs at the beginning of the document, the method in [69] analyzed only the first 20 DOM nodes to extract the features.

This aims to speed up the process but sometimes leaves out valuable information. Many keyword extraction methods use the DOM [30, 69, 70, 71, 72, 73] to extract the content of a webpage. The DOM tree represents the hierarchy of a HTML document. Filtering the content of the HTML document using various techniques yields the actual text content. The DOM forms the tree structure of the tags, which helps access textual information easily.

Structural features are language and domain independent. A structural feature exploits the clues that authors use in their texts to draw attention to important points. Writing is common in many forms in documents, which gives additional insight into the importance of words [74, 89]. A structural feature is mostly considered as a Boolean value (0: absent, 1: present), normalized by the length of the text (i.e., the presence or absence of the term at each position of the webpage). We classify these features into two groups: positional and typographical.

In web document headings, bold, italics, capitalization, and font are used to emphasize certain words and are known as typographical features. In some cases, important words are highlighted using quotation marks [57]. Positional features are associated with specific areas or HTML tags for web documents—for example, heading, title, and anchor tags—while in normal documents, important words appear at the start, end, and title.

5.1 Positional features

In an HTML document, the headings <H1> through <H6> are among the most common segments that highlight important information [11, 26, 68]. They draw users' attention to the most important parts of a document. Headings <H1> through <H3> are the most significant and are frequently used in HTML documents. Figure 10 shows an example with important parts of a webpage.

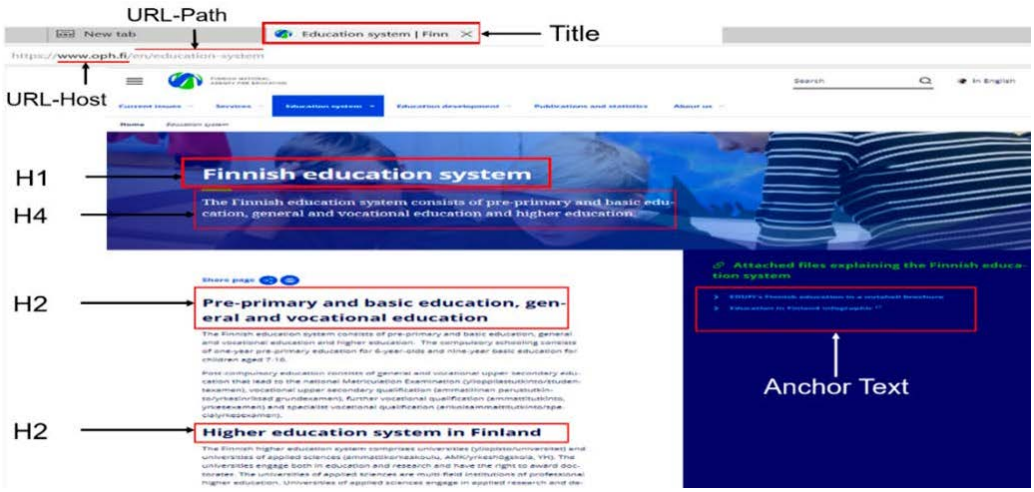

Figure 10. Title, URL, and structure of an example webpage [P2].

The title tag is an important source of information [1, 11, 50, 66, 68]. Important words are usually placed in the title tag and meta descriptions of webpages [74]. In particular, the title tag contains the text that is most important for explaining the whole document accurately. Search engine bots usually inspect the title tag of webpages for valuable information. This information assists in text processing to return relevant data to the user.

An anchor tag feature includes words that appear inside the anchor tag. Anchor text describes the target page more precisely and correctly [75]. Therefore, anchor text can be used to determine important keywords. Figure 11 shows important tags in a webpage.

```
<h1 id="page-title">BBC Homepage</h1>
<h2 >Accessibility links</h2>
<h2 class="module__title">News</h2>
<h3 class="title"> A really simple guide to the UK general election </h3>
<a href="https://www.bbc.co.uk">Home</a>
```

Figure 11. Popularity among heading and anchor tags.

Words may also appear inside span or meta tags [50, 61, 66, 68]. The meta tag contains meta information of a webpage, and typically the terms that appear in the meta tag are important. The URL feature is one of the most important features in keyword extraction. The content in URLs is usually precise and contains text that is highly condensed but relevant to the topic of the webpage [1, 66, 68]. There are three parts of a URL, as shown in Figure 12: host, path, and query [77].

Figure 12. Host, path, and query parts of URL.

In our methods [**P2, P3, P4**], the URL is divided into host and query. The host is joined with the path part of the URL. We considered each part of URL as a separate binary feature. A URL like "http://quest.edu.pk/admission-results-2022/" would have a host (i.e., "quest edu pk") and a query ("admission results 2022").

5.2 Typographical features

The most common typographical features to use in keyword extraction are underlining, bold, italics, and capitalization. Words can also be highlighted with quotation marks. Keywords can be found based on typographical emphasis. The productivity of these features depends on the type and typographic conventions of the source text, but when they are available, they are quite reliable. In [78], advertising keywords were extracted from webpages based on the assumption that capitalized words were important.

In [**P4**], we counted the occurrence of words appearing in all caps ("capital"), initial caps ("upper"), bold, or italic. The capital feature and upper feature differ from one another in that we considered initial characters of words as capitals when employing the capital feature and all letters of words as capitals when using the upper feature. The word "Car," for example, has its first character capitalized and the word "BBC" has all its letters capitalized. In Table 7, we compare the different methods using structural features.

Table 7. Comparison of methods and their use of structural features.

Method	**F1**	**F2**	**F3**	**F4**
D-rank [**P2**]	positional	[b] structure	-	Binary
WebRank [**P3**]	positional	[b] structure	-	Binary
ACI-rank [**P4**]	positional	[b] structure	[a] typographical	Binary
Yih [78]	positional	[b] structure	[a] typographical	length
Sterckx [79]	positional	[a] typographical	-	length
Turney [80]	positional	[a] typographical	-	length
Zhang [81]	positional	[c] structure	-	Binary

[a] Typographical: Word appears in all caps, initial caps, bold, or italic
[b] Structure: Word appears in URL or heading, title, or anchor tags
[c] Structure: Word appears in title, abstract, first or last sentence, or body (i.e., full text)

6 SUMMARY OF CONTRIBUTIONS

This chapter summarizes the overall contributions of our work and all four publications. The publications [**P2**], [**P3**], and [**P4**] use language-independent methods, while publication [**P1**] uses language-dependent methods. In [**P4**], we provided a systematic review of statistical, linguistic, and structural features for keyword extraction. Table 8 shows a comparison of all our proposed automatic keyword extraction methods.

Table 8. Comparison of our research methods.

	Hrank [P1]	**D-rank [P2]**	**WebRank [P3]**	**ACI-rank [P4]**
Data used	Text	Text + DOM	Text + DOM	Text + DOM
Language	English	Any	Any	Any
Preprocessing	Stem + lemma	-	-	Stem + lemma
Statistical features	TF/cluster	TF/position	TF/position	TF, DF, TF–IDF, TF–IDF Wiki-pedia
Linguistic features	Nouns + Adj + Verbs	-	-	Nouns + Adj + Verbs, Named Entity
WordNet	Synonyms	-	-	NoSynonyms
Structural features	-	H1–H6, title, anchor, URL	H1–H6, title, anchor, URL	H1– H3, title, URL, bold, italic, capital, upper
Supervised	-	-	Yes	-

We introduced an unsupervised method of keyword extraction from a single webpage in [P1] . Clustering is used to represent different concepts instead of considering each word separately. Groups of words with semantically related meanings are clustered together (e.g., Cluster–1: internet, net; Cluster–2: price, charge, cost). We selected nouns, adjectives, and verbs as candidate words and clustered them according to their semantic similarity.

During preprocessing, we applied NLP-based techniques such as filtering, cleaning text, lemmatization, stemming, and removing stopwords before clustering. We calculated semantic similarity by adopting WordNet. The top 10 clustered words were selected as keywords. We identified a good number of adjectives and verbs that could increase F-measure scores. We achieved comparable results to the term frequency, CLRank [49], and TextRank [31] methods. Hrank improved the performance of CLRank [49] after adding other POSs like adjectives and verbs (the CLRank method considers only nouns as candidate words). The combination of POSs improves performance over using only nouns.

In [**P2**], we used an unsupervised approach to extract keywords from a webpage. Through the DOM structure, we extracted important features such as headings <H1> through <H6>, title tag <title>, anchor tag <a>, and URL parts (host and path). The candidate keywords were ranked based on their position in the content after extracting their features from the DOM structure. The top 10 ranked keywords were considered the final keywords. We tested our method on a dataset of webpages in three languages: English, Finnish, and German.

In [**P3**], we presented a supervised method for extracting keywords from webpages. Paper [**P3**] is an extension of [**P2**], where we had a hard-fixed or handcrafted feature. Instead of analyzing what kind of score and features are usually used, we analyzed different classifiers to see how they performed automatically with those features.

Initially, we extracted URLs because they contain important information about keywords. Afterwards, we extracted the DOM structure of the webpage. The next step is to apply filtering and cleaning and then extract the text content. The language detection step helps to detect the language of text and remove any irrelevant words or stopwords. All sentences are converted to unigrams, and the frequency of each word is counted in a separate dictionary.

Using the DOM, the method divides the HTML page into meaningful segments and calculates a language-independent feature vector for every word. A classifier model is trained using these features, and the trained model predicts whether a candidate word is a keyword. Candidate words with the greatest likelihood of being a keyword are then selected.

We investigated the usefulness of the features on 12 datasets (news articles and service webpages) and compared different methods of classification. Random forest performed the best and provided an improvement of up to 27.89% relative to the best existing method.

In [**P4**], we analyzed the performance of each kind of feature (i.e., statistical, structural, linguistic) to discover their relative importance. One of the key results of the analysis is that stopword removal and other preprocessing steps are the most important.

Our research focused on keyword extraction from webpages. With stopword removal, simple term frequency can be calculated with reasonable results, but preprocessing is crucial. The method, called ACI-rank[**P4**], produces results that are very close to those of the supervised method [**P3**] and can possibly be enhanced further by adding more sophisticated strategies like concept graphs.

7 SUMMARY OF RESULTS

We next summarize the main results from [**P1**]- [**P4**] and compared them to the existing methods from literature. Results are obtained for four English newspaper datasets (News1), four Finnish newspaper datasets (News2), one German language dataset and Mopsi datasets; see [**P4**] for details.

7.1 Methods compared

Four existing methods are included in the comparison: *TextRank, Yake, KeyBert, CLRank.*

TextRank [31] is unsupervised graph-based method where words represent vertices, and their relationships represent edges. It analyses local word relationships within a cooccurrence window.

Yake [35] is a light wise unsupervised method using statistical text features to determine the most relevant keywords without utilizing any dictionaries, external corpora, or the size or language of the text.

KeyBert [102] is based on BERT language model. It determines subphrases within the document that most resembles the document itself using BERT embeddings and cosine similarity. The embeddings of documents are extracted using BERT to create a representation at the document level. The embeddings of n-gram words or phrases are then extracted. The most similar words or phrases are then selected based on cosine similarity.

CLRank [49] is a clustering-based method that selects nouns as candidate keywords. It uses semantic similarity to obtain clusters of the words. The importance of the clusters is determined by the distribution of its words across the webpage. Unimportant words appear only at one place of the documents whereas important words are expected to scatter throughout the page.

7.2 Evaluation measures

Classical *precision, recall* and *F-score* measures are used. They are derived from three parameters: true positive (TP), false positive (FP), and false negative (FN). *True positive* is the number of correct keywords detected. *False positive* (FP) is the number incorrect keywords detected. *False negative* (FN) is the number of correct keywords missed. They are calculated as:

$$Precision = \frac{TP}{TP + FN} \quad (4)$$

$$Recall = \frac{TP}{TP + FP} \quad (5)$$

$$F - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (6)$$

Precision is the degree of purity of information retrieval and recall refers to the degree of completeness of the retrieval. Precision and recall have the opposite relationships; when one increases the other decreases. Besides the standard precision and recall measures (hard evaluation) we also use their soft variants [84] (soft evaluation).

7.3 Results

Results in Table 9 show that among the unsupervised methods, the proposed ACI-rank works better (0.47) and is close to the supervised method WebRank (0.50) according to the soft measure [84].

While Yake is slightly better for well-structured newspaper datasets (including Finnish newspapers), the proposed method is clearly superior for the most heterogeneous Mopsi datasets. Moreover, Finnish newspaper data produced significantly worse results than English newspaper data. In KeyBert, Yake, and WebRank, significant differences were noted, but even the frequency-based baseline deteriorated despite access to Finnish and German stopwords. Due to its use of WordNet for synonyms, the Hrank [**P1**] method is limited to English-language datasets.

An example is shown in Table 11. *Toys* is the only extracted keyword that appears in the ground truth (GT) as such. Hard evaluation cannot recognize different forms of the same word. For example, *car* and *cars* are the same word but considered a failure by the hard evaluation. We therefore need the soft evaluation approach to avoid too biased evaluation procedure.

Table 9. Results using hard and soft measures.

			Hard measure		
Method	**News1**	**News2**	**Mopsi**	**German**	**Average**
TextRank [31]	0.23	0.07	0.05	0.11	0.12
Yake [35]	0.18	0.09	0.03	0.10	0.10
KeyBert [102]	0.10	0.08	0.09	0.05	0.08
CLRank [49]	0.29	0.14	0.06	0.12	0.15
Hrank [**P1**]	0.22	-	-	-	-
D-rank [**P2**]	0.30	0.13	0.12	0.21	0.19
WebRank [**P3**]	0.40	0.26	0.04	0.21	0.23
ACI-rank [**P4**]	0.33	0.16	0.14	0.18	0.20
			Soft measure		
TextRank [31]	0.44	0.36	0.19	0.37	0.34
Yake [35]	0.45	0.30	0.12	0.30	0.29
KeyBert [102]	0.28	0.16	0.14	0.11	0.16
CLRank [49]	0.49	0.39	0.19	0.37	0.36
Hrank [**P1**]	0.53	-	-	-	-
D-rank [**P2**]	0.49	0.41	0.25	0.45	0.40
WebRank [**P3**]	0.60	0.47	0.23	0.46	0.44
ACI-rank [**P4**]	0.68	0.39	0.36	0.44	0.47

Table 10. hard and soft evaluation example.

GT	Toys; Child; Car; Games; Player		
Keywords	Toys; Children; Cars; Game; Players		
Evaluation	**PR**	**Recall**	**F-Score**
Hard	0.20	0.20	0.20
Soft	0.84	0.93	0.88

8 CONCLUSION

In this thesis, we have introduced four new keyword extraction methods for webpages: Hrank, D-rank, WebRank, and ACI-rank. Among these methods Hrank is the only language-dependent method; the other three are language independent. The problem of language independence in keyword extraction has thus been addressed. All three of the language-independent methods utilize the DOM, which helps to detect the most important positions in the heterogenous structure of a webpage. Moreover, using the DOM structure, we divide those important positions into useful features. We observe that the distribution of words over DOM text nodes can be used to select keywords more effectively than term frequency in keyword extraction.

In ACI-rank, we compared the performance of the three different types of features: statistical, structural, and linguistic. Among the key findings is that stopword removal and other preprocessing steps are most important. There is no doubt that term frequency with stopword removal works reasonably well, but preprocessing plays a critical role. We also found that linguistic features help extract keywords that are more relevant to the content of the webpage.

There are two limitations of the current work. First, the goal was defined as keyword *extraction* where only words found on the web page were considered. The actual task would be to *annotate* the page with the most descriptive words, regardless of whether they appear on the web page. Second, all methods extract only single words (*keyword*), whereas many practical applications use *key phrases*. It is expected, however, that most methods developed will generalize from keywords to key phrases in a rather straightforward manner.

Future research should also be made for the following:

- Using web crawling to bridge the gap between keyword extraction and keyword assignment and annotating a webpage using keywords from similar webpages
- Finding ways to improve the WebRank method by training and testing a classifier on different datasets

- Improving the WebRank method by involves training a classifier on a variety of language datasets and testing it against them
- Studying solutions for keyword extraction from multilingual webpages so that they can be used in a more practical way
- Applying clustering based on semantic or syntactic similarity [85] instead of the simple no-synonym approach

Keyword extraction could also benefit from ideas from other summarization tasks, such as title extraction [86] and representative image selection [87], or from constructing a complete summarization covering all three tasks.

REFERENCES

[1] I. Pengqi, J. Azimi, and R. Zhang, Automatic Keywords Generation for Contextual Advertising, In Proceedings of the companion publication of the 23rd international conference on world wide web, ACM, pp. 345–346, New York, 2014.

[2] A. Usai, M. Pironti, M. Mital, and C. A. Mejri, Knowledge discovery out of text data: a systematic review via text-mining, Journal of Knowledge Management, 2018.

[3] M. Krapivin, A. Autayeu, M. Marchese, E. Blanzieri, and N. Segata, Keyphrase extraction from scientific documents: Improving machine learning approaches with natural language processing, Lecture Notes in Computer Science,6102, pp. 102–111, 2010.

[4] O. Alqaryouti, H. Khwileh, T. Farouk, A. Nabhan, and K. Shaalan, Graph based keyword extraction, In intelligent natural language processing: trends and applications, Studies in computational intelligence, Springer, Cham, vol. 740, 2018.

[5] M. Grineva, M. Grinev, and D. Lizorkin, Extracting key terms from noisy and multi-theme documents, In Proceedings of the 18th international conference on World Wide Web, ACM New York, NY, USA, pp. 661–670, 2009.

[6] K. Zhang, H. Xu, J. Tang, and J. Li, Keyword extraction using support vector machine, In international conference on webpage information management, Springer, Berlin, Heidelberg, pp. 85-96, 2006.

[7] S. Siddiqi, and A. Sharan, Keyword and keyphrase extraction techniques: a literature review, In international journal of computer applications,109 (2), pp. 18–23, 2015.

[8] N. Gali, and P. Fränti, Content-based title extraction from webpage, In WEBIST international conference, pp. 204–210, 2016.

[9] B. Armouty, and S. Tedmori, Automated Keyword Extraction using Support Vector Machine from Arabic News Documents, In IEEE Jordan international joint conference on electrical engineering and information technology (JEEIT), pp. 342–346, 2015.

[10] M. Chen, J. T. Sun, H. J. Zeng, and K. Y. Lam, A practical system for keyphrase extraction for webpages , In Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 277–278, 2005.

[11] P. Sun, L. Wang, and Q. Xia, The Keyword Extraction of Chinese Medical Webpage Based on WF TF-IDF Algorithm, In international conference on cyber enabled distributed computing and knowledge discovery (CyberC), pp.193–198, 2017.

[12] A. Hulth, Improved automatic keyword extraction given more linguistic knowledge, In Proceedings of the conference on empirical methods in natural language processing, pp. 216–223, 2003.

[13] S. Beliga, Keyword extraction: a review of methods and approaches, University of Rijeka, Department of Informatics, Rijeka, 1(9), 2014.

[14] S. Changuel, N. Labroche, and B. Bouchon-Meunier, A general learning method for automatic title extraction from HTML pages, Machine Learning and Data Mining in Pattern Recognition, pp. 704–718, 2009.

[15] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo, Extracting Semistructured Information from the Web, Stanford InfoLab, 1997.

[16] J. N. Robbins, Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics, O'Reilly Media, Inc., 2012.

[17] A. K. Mondal, D. K. Maji, and H. Karnick, Improved algorithms for keyword extraction and headline generation from unstructured text, First Journal publication from Simple groups, CLEAR Journal, 2013.

[18] A. S. Bozkir, and E. A. Sezer, layout-based computation of webpage similarity ranks, International Journal of Human-Computer Studies, vol. 110, pp. 95-114, 2018.

[19] M. Lučanský, and M. Šimko, Improving relevance of keyword extraction from the web utilizing visual style information, In International Conference on Current Trends in Theory and Practice of Computer Science, pp. 445-456, Springer, Berlin, Heidelberg, 2013.

[20] B. Carter, HTML architecture, a novel development system (HANDS): an approach for web development, In Annual Global Online Conference on Information and Computer Technology, pp. 90-95, IEEE, 2014.

[21] C. Zheng, G. He, and Z. A. Peng, Study of web information extraction technology based on beautiful soup, J. Computer, 10(6), 381-387, 2015.

[22] P. Tonella, F. Ricca, E. Pianta, and C. Girardi, Using keyword extraction for web site clustering, In Fifth IEEE International Workshop on Web Site Evolution, pp. 41-48, 2003.

[23] J. Martínez-Fernández, A. García-Serrano, P. Martínez, and J. Villena, Automatic keyword extraction for news finder, In International Workshop on Adaptive Multimedia Retrieval, pp. 99-119, Springer, Berlin, Heidelberg, 2003.

[24] A. Bellaachia, and M. Al-Dhelaan, Ne-rank: A novel graph-based keyphrase extraction in twitter, In Proceedings of IEEE CS, 2012.

[25] I. Augenstein, M. Das, S. Riedel, L. Vikraman, and A. McCallum, Extracting Keyphrases and relations from scientific publications, In Proceedings of the International Workshop on Semantic Evaluation (SemEval), Vancouver, Canada, pp. 546–555, 2017.

[26] S. Lazemi, H. E. Komleh, and N. Noroozi, PAKE: a supervised approach for Persian automatic keyword extraction using statistical features, SN Appl. Sci. 11574, 2019.

[27] A. Omar, K. Hassan, F. T. Ahmed, N. Ahmed, and S. Khaled, Graph–Based Keyword Extraction, In Intelligent Natural Language Processing: Trends and Applications, Springer, pp. 159–172, 2018.

[28] T. Jo, Keyword-based document clustering, In the proceedings of the sixth international workshop on information retrieval with Asian language, vol.11, pp.132–137, ACM, 2003.

[29] L. Page, S. Brin, R. Motwani, and T. Winograd, The PageRank citation ranking: Bringing order to the web, In International World Wide Web Conference, Brisbane, Australia, pp. 161–172, 1998.

[30] G. Salton, C. S. Yang, and C. T. Yu, A theory of term importance in automatic text analysis, In journal of the American society for Information Science, pp. 33–44, 1975.

[31] R. Mihalcea, and P. Tarau, TextRank: Bringing order into texts, In proceedings of (EMNLP04) conference on empirical methods in natural language processing, 2004.

[32] I. Witten, and G. Paynter, KEA: Practical automatic keyphrase extraction, In the proceedings of the 4th ACM conference on digital libraries, Barkeley California USA, 1999.

[33] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, YAKE! Keyword extraction from single documents using multiple local features, Information Sciences, 509, 257-289, 2020.

[34] S. Shetty, S. Akshay, S. Reddy, H. Rakesh, M. Mihir, and J. Shetty, Graph-Based Keyword Extraction for Twitter Data, In Emerging Research in Computing, Information, Communication and Applications, pp. 863–87, Springer, Singapore 2022.

[35] R. Campos, V. Mangaravite, A. Pasquali, M. A. Jorge, C. Nunes, and A. Jatowt, YAKE! Collection-Independent Automatic Keyword Extractor, In Pasi G, Piwowarski B, Azzopardi L, Hanbury A (eds) Advances in Information Retrieval, ECIR 2018, Lecture Notes in Computer Science, vol. 10772, Springer, Cham, 2018.

[36] Y. Chen, J. Wang, P. Li, and P. Guo, Single document keyword extraction via quantifying higher-order structural features of word co-occurrence graph, Computer Speech and Language, 57, 98-107, 2019.

[37] S. N. Kim, O. Medelyan, M. Y. Kan, and T. Baldwin, Automatic keyphrase extraction from scientific articles, Language resources and evaluation, 47(3), 723-742, 2013.

[38] S. Rose D. Engel N. Cramer, and W. Cowley, Automatic keyword extraction from individual documents, Text-mining: Applications and Theory, pp. 1-20 2010.

[39] T. Tomikoyo, and M. Hurst, A language model approach to keyphrase extraction, In proceedings of the ACL workshop on Multiword Expressions: Analysis, Acquisition and Treatment, vol. 18, pp. 33–40, 2003.

[40] J. K Humphreys, Phrase rate: An HTML Keyphrase Extractor, Technical Report: University of California, Riverside, pp. 1-16, 2002.

[41] J. Li, G. Jiang, A. Xu, and Y. Wang, The automatic extraction of web information based on regular expression, J. Software,12(3), 180-188, 2017.

[42] A. Awajan, Keyword Extraction from Arabic Documents using Term Equivalence Classes, In ACM transaction, vol. 14, No 2, article 7, pp. 7:1–7:18, 2015.

[43] S. Beliga, A. Meštrović, and S. Martinčić-Ipšić, An overview of graph-based keyword extraction methods and approaches, Journal of Information and Organizational Sciences, vol. 39, pp. 1-20 2015.

[44] A. Díaz-Manríquez, A. B. Ríos-Alvarado, J. H. Barrón-Zambrano, T. Y. Guerrero-Melendez, and J. C. Elizondo-Leal, An automatic document classifier system based on genetic algorithm and taxonomy, In IEEE Access, 6, 21552-21559, 2018.

[45] D. Khyani, S. B. Siddhartha, M. Niveditha, and M. B. Divya, An Interpretation of Lemmatization and Stemming in Natural Language Processing, Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology, 22, 350–357, 2020.

[46] D. R. Radev, H. Jing, M. Styś, and D. Tam, Centroid-based summarization of multiple documents, Information Processing and Management, vol. 40, pp. 919-938 2004.

[47] F. Rousseau, M. Vazirgiannis, Main core retention on graph-of-words for single-document keyword extraction, in: Advances in Information Retrieval, Springer, pp. 382–393, 2015.

[48] M. Litvak, and M. Last, Graph-based keyword extraction for single-document summarization, In Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization, Manchester, pp. 17-24, 2008.

[49] M. Rezaei, N. Gali, and P. Fränti, CLRank: A method for keyword extraction from webpages using clustering and distribution of nouns, In web intelligence and intelligent agent technology (WIIAT), IEEE/WIC/ ACM international conference, pp. 79–84, 2015.

[50] W. Zhang, W. Feng, and J. Wang, Integrating semantic relatedness and words intrinsic features for keyword extraction, In proceedings of the twenty third international joint conference on artificial intelligence (IJCAI'13), pp. 2225–2231, 2003.

[51] O. Medelyan, and I. H. Witten, Thesaurus based automatic keyphrase indexing, In ACM/IEEECS Joint Conference on Digital libraries, JCDL, pp. 296–297, 2006.
[52] D. B. Bracewell, F. Ren, and S. Kuriowa, Multilingual single document keyword extraction for information retrieval, in Natural Language Processing and Knowledge Engineering, Proceedings of 2005 IEEE International Conference, pp. 517-522, 2005.
[53] Z. Liu, P. Li, Y. Zheng, and M. Sun, Clustering to find exemplar terms for keyphrase extraction, In Proceedings of the Conference on Empirical Methods in Natural Language Processing, vol. 1, pp. 257-266, 2009.
[54] Z. Stankiewicz, and F. Hills, Systems and methods regarding keyword extraction, United States Patent publication U.S. Patent No. 8,874,568, October 2014.
[55] A. Onan, S. Korukoğlu, and H. Bulut, Ensemble of keyword extraction methods and classifiers in text classification, In Expert Systems with Applications, vol. 57, pp. 232–247, 2016.
[56] K. S. Jones, A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation, vol. 28, pp. 11-21, 1972.
[57] N. Firoozeh, A. Nazarenko, F. Alizon, and B. Daille, Keyword extraction: Issues and methods, Natural Language Engineering, 26(3), 259-291, 2020.
[58] Y. Matsuo, and M. Ishizuka, Keyword extraction from a single document using word co-occurrence statistical information, International Journal on Artificial Intelligence Tools 13(1), 157–169, 2003.
[59] W. Chung, H. Chen, and J. F. Nunamaker, Business intelligence explorer: A knowledge map framework for discovering business intelligence on the Web, In Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003.
[60] J. Xu, Q. Lu, and Z. Liu, Aggregating skip bigrams into key phrase-based vector space model for web person disambiguation, In J. Jancsary (ed), Empirical methods in natural language processing: proceedings of the conference on natural language processing, pp. 108–117, 2012.

[61] T. D. Nguyen, and M. Y. Kan, Keyphrase extraction in scientific publications, In proceedings of the 10th international conference on Asian digital libraries, pp. 317– 326, 2007.

[62] S. Momtazi, S. Khudanpur, and D. Klakow, A comparative study of word co-occurrence for term clustering in language model-based sentence retrieval, In Proceedings of Human Language Technologies (NAACL), Los Angeles, CA, USA: ACL, pp. 325–328, 2010.

[63] C. D. Maio, G. Fenza, V. Loia and M. Parente, Time aware knowledge extraction for microblog summarization on Twitter, Information Fusion Journal 28, 60–74, 2016.

[64] G. K. Palshikar, Keyword extraction from a single document using centrality measures, In Proceedings of the 2nd International Conference on Pattern Recognition and Machine Intelligence, Kolkata, India, pp. 503–510, 2007.

[65] R. Wang, W. Liu, and C. McDonald, How preprocessing affects unsupervised keyphrase extraction, In Computational Linguistics and Intelligent Text Processing, pp. 163–176, Springer Berlin Heidelberg, 2014.

[66] A. Gupta, A. Dixit, and A. K. Sharma, A novel statistical and linguistic features-based technique for keyword extraction, In international conference on information systems and computer networks (ISCON), pp. 55–59, 2014.

[67] H. Nie, Y. Yang, and D. Zeng, Keyword Generation for Sponsored Search Advertising: Balancing Coverage and Relevance, In IEEE intelligent systems, vol. 34, number 5, pp. 14–24, 2019.

[68] W. Zhang, D. Wang, D. Xue, G. Rong, and Z. Hongyuan, Advertising Keywords Recommendation for Short Text Webpages using Wikipedia, In association for computing machinery, New York, USA, vol. 3, issn 2157–6904, 2012.

[69] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma, VIPS: a vision-based page segmentation algorithm, In Microsoft technical report, MSR-TR-2003-79, 2003.

[70] F. Lei, M. Yao, and 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, Chongqing, China, pp. 323–325, 2009.
[71] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma, Extracting content structure for webpages based on visual representation, In proceedings of 5th Asia Pacific Web Conference, Xi'an China, 2003.
[72] J. Pasternak, and D. Roth, Extracting article text from the web with maximum subsequence segmentation, In proceedings of the 18th international conference on world wide web ACM, pp. 971–980, New York, NY, USA, 2009.
[73] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm, DOM-based content extraction of HTML documents, In proceedings of the 12th international conference on World Wide Web ACM, New York, NY, USA, 2003.
[74] R. M. Alguliev, and R. M. Aliguliyev, Effective Summarization Method of Text Documents, Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI'05), pp. 264–271, 2005.
[75] SEOmoz, The beginners guide to SEO, Technical report, 2012.
[76] G. Matoˇsevíc, Using anchor text to improve webpage title in process of search engine optimization, In proceedings of the Conference on Information and Intelligent Systems, Varaˇzdin, Croatia, 2015.
[77] K. L. Poola, and A. Ramanujapuram, Techniques for keyword extraction from URLs using statistical analysis, Patent Application Publication, Apr. 2, US, 2009/0089278 A1, 2009.
[78] W. -t. Yih, J. Goodman, and V. R. Carvalho, Finding advertising keywords on webpages, In Proceedings of the 15th International Conference on World Wide Web (WWW), ACM, Edinburgh, Scotland, pp. 213–222, 2006.
[79] L. Sterckx, C. Caragea, T. Demeester, and C. Develder, Supervised keyphrase extraction as positive unlabelled learning, In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, pp. 1924–1929, 2016.

[80] Turney, Learning algorithms for keyphrase extraction, Information Retrieval 2(4), pp. 303–336, 2000.
[81] C. ZHANG, Automatic keyword extraction from documents using conditional random fields, Journal of Computational Information Systems, 4.3: pp.1169-1180, 2008.
[82] C. Huang, Y. Tian, Z. Zhou, C. X. Ling, and T. Huang, Keyphrase extraction using semantic network's structure analysis, In Sixth International Conference on Data Mining (ICDM'06), pp. 275-284 IEEE, 2006.
[83] M. H. Haggag, Keyword extraction using semantic analysis. International Journal of Computer Applications, 61(1), 1-6, 2013.
[84] P. Fränti, and R. Mariescu-Istodor, Soft precision and recall. Manuscript, Software: http://cs.uef.fi/paikka/Radu/tools/SoftEval/.
[85] N. Gali, R. Mariescu–Istodor, D. Hostettler, and P. Fränti, Framework for syntactic string similarity measures, In Expert Systems with Applications, vol. 129, pp. 169–185, 2019.
[86] N. Gali R. Mariescu-Istodor and P. Fränti, Using linguistic features to automatically extract webpage title, Expert Systems with applications, vol. 79, pp. 296-312, 2017.
[87] N. Gali, A. Tabarcea, and P. Fränti, Extracting Representative Image from Webpage, In WEBIST, pp. 411– 419, 2015.
[88] P. Domingos, and M. Pazzani, On the optimality of the simple Bayesian classifier under zero–one loss, Machine learning 29 (2–3), pp. 103–130, 1997.
[89] G. Turney, Learning to extract Keyphrases from text, technical report, National Research Council, Institute for Information Technology, 1999.
[90] W. J. Wilbur, and K. Sirotkin, The automatic identification of stopwords, Journal of Information Science, vol. 18, pp. 45-55, 1992.
[91] X. Bai, and L. J. Latecki, Path similarity skeleton graph matching, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, pp. 1282-1292, 2008.
[92] G. A. Miller, and C. Fellbaum, Wordnet then and now, Language Resources and Evaluation, vol. 41, pp. 209-214, 2007.

[93] G. Salton, A. Wong, and C.-S. Yang, A vector space model for automatic indexing, Communications of the ACM, vol. 18, pp. 613-620, 1975.
[94] B. Medlock, An introduction to NLP-based textual anonymization, in Proceedings of 5th International Conference on Language Resources and Evaluation (LREC), Genes Italie, 2006.
[95] P. D. Turney, Coherent keyphrase extraction via web mining, ArXiv preprint cs/0308033, 2003.
[96] D. Kelleher, and S. Luz, Automatic hypertext keyphrase detection, In IJCAI-05, 2005.
[97] M. Litvak, M. Last, H. Aizenman, I. Gobits, and A. Kandel, DegExt—A language independent graph-based keyphrase extractor, In Advances in Intelligent Web Mastering–3: Springer, pp. 121-130, 2011.
[98] C. Florescu, and C. Caragea, Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents, In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1105-1115, 2017.
[99] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, Domain-specific keyphrase extraction, In 16th International joint conference on artificial intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, vol. 2, pp. 668-673, 1999.
[100] S. Brin, and L. E. Page, The anatomy of a large-scale hypertextual web search engine, Computer Network, and ISDN system, 30(1-7), 107–117, 1998.
[101] Y. Xue, Y. Hu, G. Xin, R. Song, S. Shi, Y. Cao, C. Lin, and H. Li, Webpage title extraction and its application, In Journal of Information Processing and Management, vol. 4, issue 5, pp. 1332–1347, 2007.
[102] M. Grootendorst, KeyBERT: minimal keyword extraction with BERT, Software: https://github.com/MaartenGr/KeyBERT, 2020.

ORIGINAL PUBLICATIONS

PAPER 1

H. Shah, M. U. S. Khan, and P. Fränti,
“Hrank: a keywords extraction method from webpages using POS tags”,
IEEE International Conference on Industrial Informatics (INDIN),
Helsinki, IEEE, 2019.
http://doi.org/10.1109/INDIN41052.2019.8972331

PAPER 2

H. Shah, M. Rezaei, and P. Fränti,
“DOM-based keyword extraction from webpages ”,
In proceedings of international conference on artificial intelligence, information processing and cloud computing (AIIPCC),
Sanya, China, Article No. 62, ACM, 2019.
https://doi.org/10.1145/3371425.3371495

PAPER 3

H. Shah, R. Mariescu-Istodor, P. Fränti,
“WebRank: Language independent Extraction of Keywords from Webpages”,
In IEEE International Conference on Progress in Informatics and Computing (PIC), IEEE, 2021.
http://doi.org/10.1109/PIC53636.2021.9687047

PAPER 4

H. Shah, P. Fränti,

“Combining statistical, structural, and linguistic features for keyword extraction from webpages Applied Computing and Intelligence, 2(2), 115-132, 2022.

http://doi.org/10.3934/aci.2022007

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 Fränti*

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.

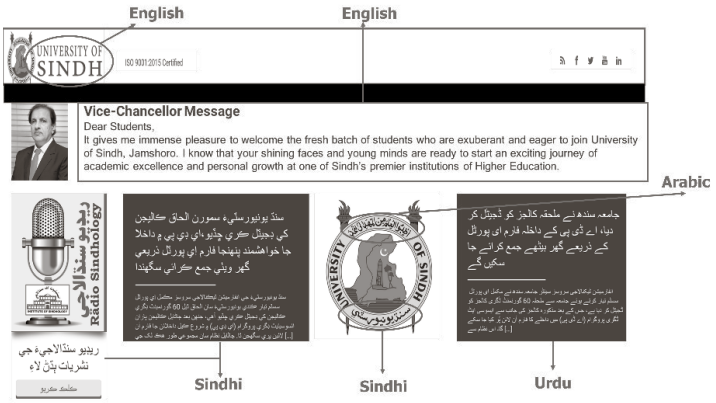

Figure 1. Example of complex multi-lingual web page with heterogenous structure.

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.

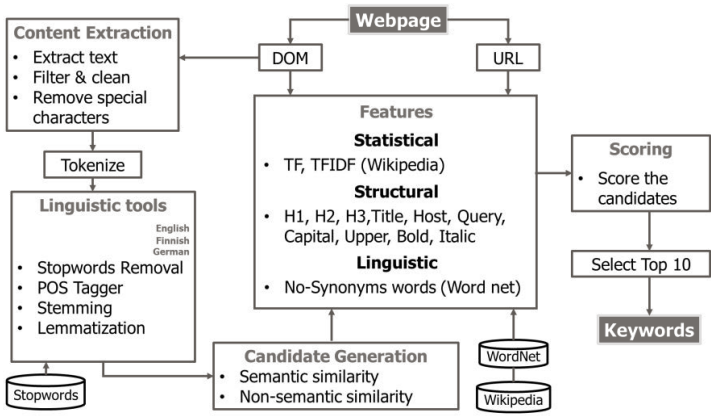

Figure 2. Framework for the keyword extraction combining statistical, structural and linguistic features.

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 naïve, 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.

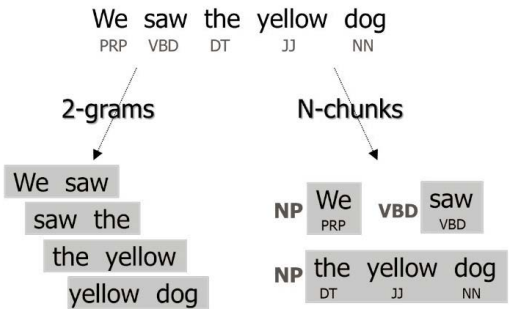

Figure 3. Examples of the N-grams (n = 2) and N-chunks (n is variable length) approaches.

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>, , <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.

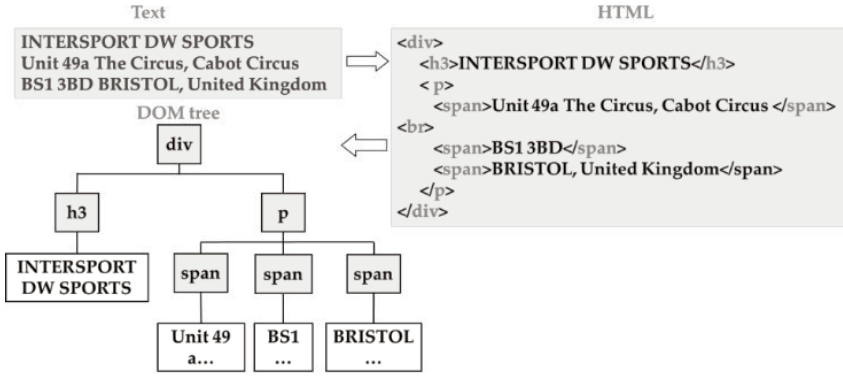

Figure 5. Example of a piece of HTML code and the corresponding DOM tree.

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>	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*:

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

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* 17*th 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* 10*th 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* 23*rd 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* 18*th 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

AIMS Press

HIMAT SHAH

Text documents are often summarized using keywords. A human can manually annotate them, or a computer can generate them automatically. Webpage automatic keyword extraction involves selecting a set of words that best describe the content of the webpage. Most existing keyword extraction methods rely on language-dependent Natural Language Processing techniques which makes it difficult to generalize the method to other languages. This research aims to find a method that can be applied to webpages regardless of their language, by extracting only language-independent features.

UNIVERSITY OF
EASTERN FINLAND

uef.fi

PUBLICATIONS OF
THE UNIVERSITY OF EASTERN FINLAND
Dissertations in Forestry and Natural Sciences

ISBN 978-952-61-4687-4
ISSN 1798-5668