WebRank: Language-Independent Extraction of Keywords from Webpages

Himat Shah, Radu Mariescu-Istodor, Pasi Fränti
School of Computing University of Eastern Finland, Joensuu Finland
himat@cs.uef.fi, radum@cs.uef.fi, franti@cs.uef.fi

***Abstract*—We present a supervised method for keyword extraction from webpages. The method divides the HTML page into meaningful segments using document object model (DOM) and calculates a language independent feature vector for each word. Based on these, we generate a classification model that gives a likelihood for a word to be a keyword. The most likely words are then selected. We analyze the usefulness of the features on different datasets (news articles and service web pages) and compare different classification methods for the task. Results show that random forest performs best and provides up to 27.8 %-unit improvement compared to the best existing method.**

Keywords—keyword extraction, DOM, Language independent

I. INTRODUCTION

The World Wide Web creates over 20,000 gigabytes of data every second1. Finding relevant and useful information from such a huge amount of data is challenging. Search engines have made the search of information much easier but finding the most relevant documents still depends on the quality of the keywords [1].

A keyword is a single word, or a sequence of words (key phrase) in the text that provide concise, high-level description of the content to readers [2]. Using keywords, it is possible to manage and classify web documents [3]. For example, a web document about portable computer manufacturer might be categorized under the keyword laptop. In applications for text mining and natural language processing (NLP), keyword extraction is a basic step used in text summarization, information retrieval, topic modelling [4], clustering [5, 23, 24] and content-based advertisement systems [6]. Keywords or key phrases are used interchangeably but researchers typically define the keyword to mean a single word and the key phrase to a sequence of words.

Keyword extraction is more difficult from webpages than from plain text. There are two main challenges. The first one is noisy and irrelevant data such as navigational bars, menus, comments and even ads (see Figure 1). The second one is the presence of multiple topics and even multiple languages [2]. Several methodologies for automatic keyword extraction have been proposed in literature but they focus on simple text and less attention is given to the webpage structure.

In this paper, we propose WebRank, a keyword extraction method specifically tailored to work on webpages by using features extracted from the Document Object Model (DOM) [7]. We used features like if a word is part of the URL, in the title, header tags or hyperlinks. Based on these features, we train a classifier to determine whether a certain word is a keyword. We experiment using different classifiers and perform quantitative and qualitative analyses to evaluate the performance and usefulness of these features. Our method is supervised and language independent except the need for stop word list.

Fig. 1. Advertisement and other irrelevant components on webpage that can disturb the keyword extraction

The remainder of this paper is organized as follows: Section 2 discusses related work, Section 3 explains the selected model and algorithm, Section 4 discusses the datasets, and Section 5 and 6 summarize and discuss the results.

II. RELATED WORK

In literature, there are numerous methods for keyword extraction. We classify these methods [7, 20, 22, 23, 24] into: (1) keyword extraction from normal text, and (2) keyword extraction from the text content of a webpage. In this paper, we focus on the second.

Most existing methods [15, 16, 17, 20, 27] from the past few decades are language dependent. Language independent approaches have been less studied because they usually perform worse than methods that utilize linguistic features. However, their main drawback is the limitation to certain pre-defined languages. Language models may not be freely available for all languages and when they do, they can often have different representations. Language independent methods would be much easier to implement in practice.

[1] https://www.webfx.com/internet-real-time

Language dependent methods commonly apply pre-processing steps that use linguistic information. Typical methods used include natural language processing (NLP) such as stemming and lemmatization for text normalization. For example, nouns like buses, sporting, and hotels normalize to bus, sport, and hotel when using stemming.

After pre-processing, keyword extraction methods apply linguistic dictionaries thesaurus such as WordNet [7, 8] and Wikipedia [2, 16] to find semantic relationship among the words. For example, internet and net are semantically the same and can be used interchangeably. These thesauruses are beneficial to enhance the quality of the keyword extraction method. However, use of these language dependent components makes it difficult to generalize the keyword extraction methods to other languages.

The method in [2] is a language independent method, which extracts the content from news webpages using structural properties and visual presentation information from CSS, such as font size and colour. A classifier is trained using these features.

Webpage segmentation is also utilized often to select the keywords. Many approaches use document object model (DOM) [9,10,11,12] to divide HTML into segments. In [7], important segments are detected and separated from the main document and candidate words is assigned scores per based on their positions and importance within the segment. Top 10 scored candidates are selected as keywords.

The method in [13] analyses only the first twenty DOM nodes to extract the features using the assumption that the most important information is in the beginning of the document. This is motivated also by speedup of the process. However, this sometimes misses valuable pieces of text [8]. A variant called VIPS [8] utilizes the content in more balanced manner.

Machine learning plays an important role in both types of keyword extraction. Machine learning is divided into supervised and unsupervised learning. In supervised learning, the system is trained using ground truth, which are web documents and the list of expected keywords.

A method called KEA [15] is one of the best-known supervised methods in which the classifier is trained using the naive Bayes learning algorithm. Three input features are extracted: term frequency-inverse document frequency (TFIDF), distance of candidate phrase in text from beginning of the text, and the frequency of the key phrase. Each phrase is considered separately from other phrases by simple statistical analysis. KEA++ [16] is an improved variant that uses semantic information from Wikipedia. Another extension of KEA is the keyphind system [17] which improves the performance using digital libraries and so-called keyphind indexing. These indexes are much smaller than normal full-text index, which makes it easy to implement and efficient. The method in [4] uses regression model trained on a set of human-labelled keywords.

In [18], two methods for title extraction are proposed by utilizing the HTML content. A classifier is trained using DOM-based features. The first method extracts features such as font, tag, format information and while the second method focuses on page layout, block, and unit position. Different machine learning classifiers were compared, and conditional random fields (CRF) was shown to outperform SVM. GenEx [27] is another supervised method applying a set of heuristic rules for training the corpus by a genetic algorithm. In [19], statistical language models using pointwise KL-divergence is used to score important phrases. High precision score was achieved but two problem were reported; it requires long training time, and it is language dependent.

Unsupervised machine learning has also been used for the keyword extraction task including graph-based and clustering approaches [20, 37]. TextRank [20] is a graph-based algorithm where words are vertices, and their relationship are edges. It is unsupervised language-dependent method which analyses relationship between words locally within so-called co-occurrence window.

Yake [21] is a lightweight, unsupervised, automatic keyword extraction method. Yake method uses statistical text features extracted from an individual document to determine the most relevant keywords. Yake does not require training on a specific set of documents, nor do it depend on dictionaries, external corpora, text size, language, or domain.

KeyBert [22] is a simple and easy to use keyword extraction method that leverages BERT language model. It uses BERT embeddings and cosine similarity to determine which sub-phrases in the document are most similar to the document itself. First, the embeddings of documents are extracted using BERT to create a document-level representation. Word embeddings are then extracted for n-gram words or phrases. At the end, most similar words or phrases are chosen according to the cosine similarity.

In [23], only noun phrases are used as candidate keywords. Other parts of speech such as adjectives and verb were also allowed in [24]. After extracting the parts of speech candidate keywords, clustering based on semantic relations is applied where top-ranked clusters are selected as the source for the keywords. Other language independent methods in literature include Rake [25], Drank [7] and DegExt [27].

Method [28] is a graph-based keyword extraction method. The experiment applied on corpus of Medline scientific abstract, multiple variants of a graph called 3-graphlets and 4-graph produced. Next, Naive Bayes classifier trained to decide whether a word is a keyword or not. The method achieved significant result over TF-IDF and baseline methods.

PageRank [29] takes text as input from single or multi-text documents. It first combines the multiple text sources and then splits the text into sentences and convert the sentences into vector embeddings. After that it calculates the similarities between the vectors and stores them into a similarity matrix, which is converted into graph where sentences are the vertices, and the edges are weighted by the similarity score.

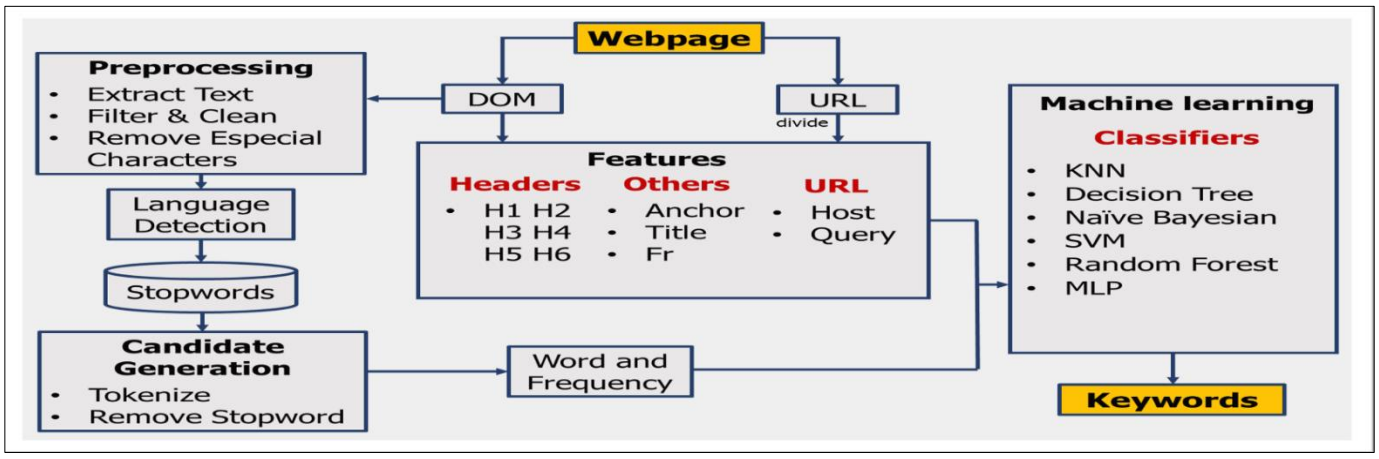

Fig. 2. Workflow architecture of WebRank.

III. KEYWORD EXTRACTION ALGORITHM

Workflow of the proposed WebRank method is shown in Figure 2. The method has four modules: 1) preprocessing, 2) candidate generation, 3) feature extraction, 4) classifier training. We discuss each module separately as follow.

A. Preprocessing

In this module, we first extract the text of the given webpage and then apply different natural language processing (NLP) based techniques to clean and filter the text. In the beginning, content of hypertext mark-up language (HTML) is downloaded using domain object model (DOM). DOM is a powerful tool and easy to implement. It forms tree structure of the HTML tags that makes easier to access the content of the web page [8]. The same downloaded content and the URL of the page are also given to the feature extraction module.

Next, text cleaning and filtering functions are applied to the content. Filtering function filters out text that is part of cascade style sheet (CSS) and java script (JS). In addition to HTML components, web pages also often contain CSS and JS components. CSS components specify a page's style, layout, and format, but do not provide any useful information for keyword extraction task. Instead, they specify how HTML elements should be displayed. In web pages, Java script is used to make them interactive. It can, for instance, validate email addresses entered by users in a form field.

We trim them out because they are mainly used for the structure of the webpage which rarely contains useful content-related words. Special characters (such as #, &, %) are also removed.

B. Candidate Generation

After preprocessing, text is tokenized into unigram tokens. A token is a white space-separated words that is treated as a unit of text [23]. We consider every unique token as a candidate keyword. In typical normalization process, the words would be stemmed or lemmatized to their root form from inflected form as in Drank [7]. For example, word cars after applying stemming function becomes car. But, sometimes stemming leaves unmeaningful words such as news as a new. We omit this step because it requires language models, and we want to avoid excessive dependency on the language.

However, we do remove tokens that are recognized as stopwords, which are frequent words (for, the, and) that appear in almost any text. This makes the method semi-dependent to language. We argue that stop word lists are widely available with limited resources from Wikipedia for example. This requires language detection function, which provides the name of the language; and a list of the stopwords extracted from python language small libraries. The method removes stopwords of both the detected language and English language. This is because English language stopwords are commonly present in the HTML of all webpages.

After these processing steps, we count frequencies of all remaining candidate words.

C. Feature extraction

For each candidate word, we extract the following features:

1. Frequency (number of distinct occurrences in document)
2. H1 (binary feature specifying if word appears in <h1> tag)
3. H2 (appears in <h2> tag)
4. H3 (appears in <h3> tag)
5. H4 (appears in <h4> tag)
6. H5 (appears in <h5> tag)
7. H6 (appears in <h6> tag)
8. Anchor (appears in <a> tag)
9. Title (appears in <title> tag)
10. URL Host (appears in the host part of the URL)
11. URL Query (appears later in the URL)
12. Page size (total number of words in document)

The last feature is the same value for every word, and it acts merely as a scaling factor for the frequency feature. For example, a frequency of 5 can be considered more important when document has 100 words but less important when the document has 10,000 words. In Drank [7], the features are weighted to compute a score for each word as follows:

$$Score = \underline{0.2}f_1 + 6f_2 + 5f_3 + 4f_4 + 2f_5 + 2f_6 + 2f_7 + 2f_8 + 5f_9 + 5f_{10} + 4f_{11} \quad (1)$$

if $f_{12} \leq 50$ and,

$$Score = \underline{0.5}f_1 + 6f_2 + 5f_3 + 4f_4 + 2f_5 + 2f_6 + 2f_7 + 2f_8 + 5f_9 + 5f_{10} + 4f_{11} \quad (2)$$

otherwise, where f_{1-12} are the feature values. The highest scoring words are considered as the keywords. The number of keywords to be extracted depends on properties of the dataset and the ground truth information, agreed here as 10 for Mac, 10 for Guardian and 5 for Mopsi Services (see Section IV).

D. Classifiers

In the proposed method, the weights are optimized in the training. We consider six alternative classifiers for this:

- KNN [30]
- Decision tree [31]
- Naïve Bayesian [32]
- SVM [33]
- Random forest [34]
- MLP [35]

We next compare the different classification methods. To test, we performed 5-fold cross validation, where each dataset was divided into 5 equal parts and 80% of the data was used to train and 20% to test. We optimized the decision tree using the entropy criterion and used the best splitting choice at each step. We found that k = 3 gives the best F-score for KNN [27] after which the quality degrades steadily. Default parameters are used whenever possible as we want to have comparable results to the existing methods without overfitting the models too much.

IV. EXPERIMENTS

To evaluate the performance of the methods, we carry out experiments on twelve publicly available datasets. The detail description of these datasets is given in Table I. There are 2936 unique webpages present in these datasets. Examples are shown in Figure 3. The contents of the datasets appear in three languages: English, Finnish, and German. These datasets can be found via the following links2. The number of keywords to be extracted depends on the properties of the dataset. The ground truth is set to 5 in case of Mopsi Services, and 10 in case of the Newspaper datasets.

We analyse the performance of individual dataset and the behaviour of different keyword extraction methods when applied to these datasets. The goal is to find out why one dataset is easier than another one; and second, what makes a set easier or more difficult. We study three main aspects: (1) importance of the features; (2) how easy is a dataset; (3) other factors effecting the performance of a method. To find the answers we collect numerical results and calculate average accuracy compared to the ground truth (GT).

A. Evaluation measures

Hard evaluation comprises of three classical measures: precision, recall and F-score. These measures are generally used for evaluating keyword extraction methods. The scores calculated by these measures are based on three parameters: (1) true positive (TP) is the number of detected keywords that appear also in the ground truth; (2) false positive (FP) is the number of detected keywords that are not in GT; (3) false negative (FN) is the number of ground truth keywords that were missed by the method. In hard evaluation, the correctness of a keyword requires that it is an exact match to a keyword in the ground truth. Using these three parameters, the measures are calculated using the formulas given in Table II.

TABLE I. DATASETS USED IN THE EXPERIMENTS

Dataset	Data Source	Web pages	Keywords (avg)
English			
Guardian	theguardian.com	421	13.4
Herald	universityherald.com	300	9.0
Indian	indianexpress.com	329	6.1
Mac	macworld.com	204	7.5
Finnish			
Kaksplus	kaksplus.fi	200	5.4
Kotiliesi	kotiliesi.fi	210	6.5
Ruoka	ruoka.fi	200	7.4
Taloussanomat	taloussanomat.fi	210	9.8
Urheilulehti	urheilulehti.fi	200	6.6
Uusisuomi	uusisuomi.fi	200	10.8
German			
German	multiple URLs	81	16.2
English & Finnish			
Mopsi	multiple URLs	381	2.5

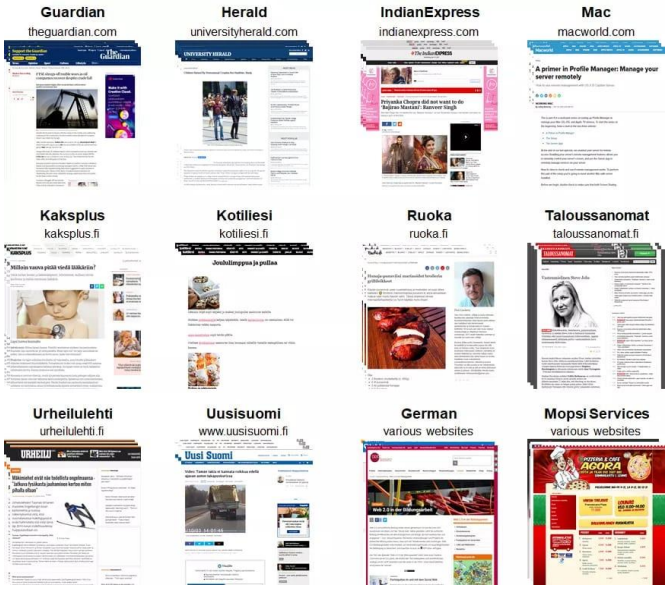

Fig. 3. Examples of webpages in the datasets.

[2] http://cs.uef.fi/mopsi/data

The problem of using the hard evaluation is that it almost always provided very low score even when the method would perform reasonably well by subjective evaluation. The reason is the requirement of the exact match. For example, if the ground truth has word student but the extracted keyword is students, the hard evaluation would count it as an incorrect choice despite the two words are practically the same.

To provide better evaluation, we therefore use also soft evaluation presented in [36]. In soft evaluation, two words are compared based on some similarity measure. Instead of using binary matching, we calculate soft precision and soft recall values based on the similarity score. We use Jaccard at the token level and 3-gram with padding at the character level based on its good performance in [37]. We can see clear effect of using the soft evaluation over the hard evaluation. For example, the word student and students are almost identical. While completely ignored by the hard evaluation, soft evaluation of these two words gives score 0.74. Soft evaluation is especially useful when dealing with inaccurate human annotated ground truth.

TABLE II. HARD EVALUATION MEASURES USED.

$$Precision = \frac{TP}{TP+FN} \quad (3)$$

$$Recall = \frac{TP}{TP+FP} \quad (4)$$

$$F-Score = 2 \times \frac{precision \times recall}{precision + recall} \quad (5)$$

B. Feature Importance

Features 2, 9 and 11 (H1, title tags, query part of the URL) appear to be significant in case of all datasets. Feature 11 is important especially for the news articles because URL contain important information. For example, MACWorld pages are formatted as: https://www.macworld.com/article/3512017/how-to-use-apple-id-to-create-passwords.html. In case of Mopsi services, URL is helpful only when the service does not have its own web page but are listed as a part of a larger service directory or is a Facebook page: https://www.facebook.com/kotipizza. If the service has its own web page, Feature 10 becomes more important. For example: https://www.kotipizza.fi.

In Herald dataset, ground truth keywords area present in the query part of URL in most cases. That is the reason why it becomes easiest dataset and provide highest f-score. Important thing to note about Features 2 and 9 is that their contents are almost similar, and they lead to highest percentages of GT keywords found in these datasets. This makes sense for news articles because they have usually long, descriptive titles emphasized by H1 and title tags (the latter is visible only in the browser tab). This feature is important also for Mopsi services because the title of the services often contains descriptive part: Pizza Express, Café Manta, Ravintola Riemuralli (Ravintola means Restaurant in Finnish language). To sum up, these three features are powerful, and what makes these datasets easy.

Feature 8 (the anchor tag) is significant only in case of the news articles, presumably because they can have links to related information. Using hyperlinks for Mopsi services can even be harmful as links to other similar services is usually not present because it would mean linking to your competitor. The features 6, 7 and 10 (H5, H6, URL-Host) are less significant and have zero frequencies in case of news articles datasets.

In Feature 10, there appears repetition of the same words in all the webpage even when none exists in the GT keywords. For example, in Herald dataset, word universityherald word appears 300 times but not even once in the ground truth. However, separated words like university and herald appear many times. Features 6 and 7 are not present at all in 6 out of 12 datasets, and infrequently in the rest of the 6 datasets. Feature 5 (H4) is missing from three of the English datasets and found only in the second dataset.

In general, English datasets are easier (3 out of 4 have top-3 median scores). Ruoka.fi is the only exception among the Finnish datasets scoring among top-3. Herald is the easiest dataset. It provides the highest f-score (0.70) among all the other datasets. The reason why it is easy is because the ground truth (GT) keywords are also attached in the query part of URL. Any method that utilizes this fact is likely to perform well.

The hardest dataset, by far, is the Mopsi Services. While the newspaper data are systematically created, Mopsi services represent a wide range of services collected by crowdsourcing with human annotated ground truth keywords. These web pages are heterogenous and lack uniform structure and they are sometimes in English, sometimes in Finnish, and sometimes in mixed languages of these two. The human annotated keywords also do not follow any systematic rules. Because of these reasons, the dataset provides a good challenge and can cause problems for methods that are heavily based on linguistic features.

C. Affecting factors

We can list factors that improve or degrade the performance of the keyword extraction methods when applied on dataset as follows: (1) Number of words in the page on average; (2) number of keywords in the GT; (3) GT keywords that do not appear in the webpage; (4) stopwords used as GT keywords. Table III shows statistics about these factors.

TABLE III. PROPERTIES OF THE DATASETS.

Dataset	Words (avg)	GT keywords	GT not in page	Stop word in GT
Guardian	1224	**5637**	**692**	**413**
Herald	1038	2698	268	56
Indianexpress	**1438**	2023	128	28
MACWorld	1254	1531	38	21
Kaksplus	804	1077	46	7
Kotiliesi	735	1353	68	4
Ruoka	422	1480	37	16
Taloussanomat	1875	2053	119	14
Urheilulehti	1062	1314	142	4
Uusisuomi	**2192**	**2178**	**180**	14
German	1180	**1391**	**666**	**87**
Mopsi	460	952	**607**	1

From Table I we can see that German dataset has most ground truth keywords (16.2) assigned, on average, and Guardian has the 2nd most (13.4). Mopsi services has the least (2.5). The first two have also a lot of annotated GT keywords that do not exist in the webpage (692 and 666). For this reason, most methods provide poor f-scores for these two datasets. Most methods are limited to extract only existing words from the webpage without any attempt to summarize a set of words. Another reason is that the webpages contain stopwords. For example, and is the most frequent stop word in GT, which also indicates that key phrases are used in addition to keywords.

TABLE IV. KEYWORDS EXTRACTED BY DIFFERENT METHODS FROM GUARDIAN WEBPAGE.

Ground Truth (GT)
BG Hellenic Bottling Company Amec Tullow Oil BT Wood Group Weir Royal Dutch Shell Randgold Resources
D-rank
Oil Companies Rouble Crude Ftse Despite Fall Shrugs Woes Recover
TextRank
Oil Guardian Companies Group UK Data Edition Russia Sign Business Home Figures
Yake
Share Oil Edition Comments Switch Business Markets Guardian Figures UK
KeyBert
Email Markets Football Lifestyle Switch Edition Guardian Home Share Oil Business Companies UK Share
WebRank / Decision Tree (DT)
Oil Guardian Sign Football Comments
WebRank / KNN
Guardian UK Markets Rate Reserved
WebRank / SVM
Business Guardian Change Edition Comments
WebRank / MLP
Business Guardian Switch UK Share
WebRank / Random Forest
Oil Edition Interest Points Reserved
WebRank / Naive Bayes
Change Edition Topics View Comments

D. Performance of the other methods

We compare the performance of WebRank with other methods including Drank [7], TextRank [20], Yake [21] and KeyBert [22]. We have implemented all the methods by ourselves. WebRank restricts to only unigram words as a keyword. It also removes short words like os, th and xo, during the preprocessing step whereas D-rank can select short words of only two character. Yake and TextRank also allow bigrams and trigrams as keywords as well as short words of two characters. They therefore perform well when the dataset has key phrases as ground truth keywords. Biggest limitation of TextRank is that it utilizes only the text content of a webpage and ignores any DOM-related and visual features like header tags.

Table IV shows a qualitative example of keywords extracted by the methods. TextRank and Yake extract keywords in all languages, but stopwords and common words found other than English language. For example, from German language webpage. Out of extracted 12 words by TextRank, 11 are German language stopwords. D-rank generate better result as three out of ten keywords are present in ground truth and no stopwords are selected.

E. Soft and Hard evaluation

Table V shows an example how the result by soft evaluation differs from that of the hard evaluation. Overall, there is no big difference in ranking of the different methods when evaluated by the hard versus the soft measure. Hard evaluation is too rough as it penalizes even minor spelling differences. The soft measure is more realistic, and it can make distinction between the seemingly similarly performing methods and have scores at the more appropriate scale instead of close to 0.

TABLE V. COMPARING HARD AND SOFT EVALUATION

GT	student, university, tuition, opportunities		
Extracted	students, university, lecture, chance, free		
Evaluation	**Precision**	**Recall**	**F-score**
Hard	0.20	0.25	0.22
Soft	0.46	0.56	0.50

Table VI shows performance evaluation of all keyword extraction methods with all datasets. The first 6 methods are all new ones proposed in this paper. Among the different classifiers, Random Forest produced the highest average score according to the hard evaluation, and 2nd highest in the soft evaluation. Naïve Bayes does not produce best score for any dataset. All other classifiers are useful at least once. On average, the proposed method provides higher average score than any of the existing methods regardless which classifier was used.

Table VII demonstrates the advantage of soft versus hard evaluation. Delta is the difference from hard to soft calculated in percentage. Mopsi datasets and German datasets show notable differences. Hard and soft changes in the Herald dataset are minimal compared with other datasets.

The language has some effect on the performance but there is no big difference between the methods. According to the soft evaluation, SVM (0.49) is only slightly better than Random Forest (0.48) and MLP (0.48). Decision tree performs better with the English datasets (Guardian, Herald, Indian, Mac). Among the other methods, Text Rank performance slightly better and Yake slightly worse with English language compared to Finnish. KeyBert has poor performance in all cases.

Mopsi is the most challenging dataset used due to its irregular and multilingual content. The supervised methods were not able to give significant improvement over their unsupervised counterpart, D-rank. The best results were as low as 0.26, which shows that there is still room for improvements.

TABLE VI. PERFORMANCE EVALUATION (F-SCORES) OF ALL METHODS WITH ALL DATASETS. THE BEST METHOD IS EMPHASIZED BY BLUE, AND ALL THE EXISTING METHODS BY GRAY BACKGROUND

HARD EVALUATION											
Dataset	**Dec. Tree**	**Rand. Forest**	**KNN**	**SVM**	**MLP**	**Bayes**	**D-rank**	**Text Rank**	**Yake**	**KeyBert**	**Max**
Guardian	0.24	0.26	0.24	**0.32**	0.22	0.18	0.18	0.19	0.12	0.06	**0.32**
Herald	0.64	0.68	0.67	**0.70**	0.69	0.58	0.49	0.25	0.33	0.19	**0.70**
Indian	0.28	**0.36**	0.31	0.31	0.33	0.26	0.31	0.23	0.08	0.03	**0.36**
Mac	0.23	0.29	0.26	0.31	**0.35**	0.24	0.22	0.24	0.20	0.10	**0.35**
Kaksplus	0.27	**0.29**	0.21	0.24	0.22	0.16	0.19	0.07	0.16	0.01	**0.29**
Kotiliesi	0.22	0.23	0.19	0.23	**0.23**	0.22	0.16	0.07	0.13	0.06	**0.23**
Ruoka	0.35	0.39	0.28	0.28	**0.40**	0.34	0.18	0.13	0.12	0.03	**0.40**
Taloussanomat	**0.20**	0.19	0.17	0.12	0.03	0.07	0.08	0.08	0.06	0.09	**0.20**
Urheilulehti	0.21	0.25	0.23	**0.30**	0.09	0.16	0.09	0.14	0.15	0.04	**0.30**
Uusisuomi	0.21	**0.23**	0.18	0.20	0.13	0.11	0.08	0.05	0.09	0.08	**0.23**
German	0.11	0.13	0.11	0.19	**0.21**	0.19	**0.21**	0.11	0.10	0.05	**0.21**
Mopsi	0.03	0.04	0.04	0.09	**0.11**	0.09	**0.12**	0.06	0.03	0.09	**0.12**
Average	0.25	**0.28**	0.24	0.27	0.25	0.22	0.19	0.14	0.14	0.06	**0.28**
SOFT EVALUATION											
Guardian	0.46	0.49	0.46	**0.54**	0.46	0.45	0.41	0.44	0.43	0.14	**0.54**
Herald	0.77	0.80	0.79	**0.83**	0.81	0.72	0.64	0.46	0.71	0.41	**0.83**
Indian	0.43	**0.49**	0.45	0.46	0.49	0.43	0.45	0.39	0.26	0.19	**0.49**
Mac	0.48	0.54	0.51	0.55	**0.59**	0.48	0.44	0.46	0.39	0.39	**0.59**
Kaksplus	0.45	0.47	0.42	0.46	**0.47**	0.41	0.44	0.32	0.44	0.10	**0.47**
Kotiliesi	0.45	0.46	0.43	0.50	**0.54**	0.54	0.47	0.39	0.42	0.13	**0.54**
Ruoka	0.57	0.60	0.53	0.53	**0.64**	0.61	0.50	0.43	0.22	0.20	**0.64**
Taloussanomat	**0.40**	0.40	0.38	0.35	0.26	0.32	0.30	0.32	0.15	0.20	**0.40**
Urheilulehti	0.43	0.46	0.43	**0.51**	0.30	0.40	0.34	0.39	0.38	0.19	**0.51**
Uusisuomi	0.46	**0.47**	0.43	0.47	0.43	0.41	0.38	0.35	0.24	0.16	**0.47**
German	0.40	0.39	0.38	0.46	**0.47**	0.44	0.45	0.37	0.30	0.14	**0.47**
Mopsi	0.17	0.17	0.17	0.23	**0.26**	0.24	0.25	0.19	0.12	0.11	**0.26**
Average	0.46	0.48	0.45	**0.49**	0.48	0.45	0.42	0.38	0.33	0.19	**0.48**

TABLE VII. SOFT VS. HARD EVALUATION OF RANDOM FOREST

Dataset	**Hard**	**Soft**	**Delta**	**Av. Delta**
Guardian	0.26	0.49	85 %	**55%**
Herald	0.68	0.80	19 %	
Indian	0.36	0.49	37 %	
Mac	0.29	0.54	87 %	
Kaksplus	0.29	0.47	62 %	**86%**
Kotiliesi	0.23	0.46	102 %	
Ruoka	0.39	0.60	54 %	
Taloussanomat	0.19	0.40	111 %	
Urheilulehti	0.25	0.46	80 %	
Uusisuomi	0.23	0.47	108 %	
German	0.13	0.39	208 %	**263%**
Mopsi	0.04	0.17	317 %	
Average	0.28	0.48	72 %	-

V. CONCLUSION AND FUTURE WORK

We propose a supervised method for keyword extraction trained using DOM-based features. Unlike many existing methods, the proposed method does not rely on complex NLP components. This makes it fast and simple in implementation with only weak dependence on language.

We test the method on twelve corpora in three different languages using both hard and soft evaluation. Soft evaluation provides a chance to evaluate the method accurately and weight candidate words in a flexible way compared to the hard evaluation. The results show significant improvement of the best previous DOM-based method (D-rank). On average, we achieve f-score of 0.28 (hard evaluation) and 0.49 unit (soft evaluation) compared to 0.19 and 0.42 of D-rank.

In contrast to existing methods, the proposed method can extract keywords of multiple languages including English, Finnish, and German. It is suitable for processing of a single text and multi-text. A drawback of the algorithm is that it fails to find whether the valuable words will affect the issue of adjacent nodes weight transfer. Better analysis of the importance of the features is a point of future research.

REFERENCES

[1] P. Sun, L. Wang, and Q. Xia , "The Keyword Extraction of Chinese Medical Web Page Based on WF-TF-IDF Algorithm," In International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 193–198, 2017.

[2] M. Grineva, M. Grinev, and D. Lizorkin, "Extracting key terms from noisy and multi-theme documents," In ACM International Conference on World Wide Web, ACM New York, NY, USA, pp. 661–670, 2009.

[3] S. Lazemi, H. Ebrahimpour-Komleh, and N. Noroozi, "PAKE: a supervised approach for Persian automatic keyword extraction using statistical features," SN Appl. Sci. 1, 1574, 2019.

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

[5] P. Tonella, F. Ricca, E. Pianta, and C. Girardi, "Using keyword extraction for website clustering," In proceedings of fifth IEEE International Workshop on Web Site Evolution, pp. 41–48, 2003.

[6] Z. Stankiewicz, and F. Hills, "Systems and methods regarding keyword extraction," United States Patent publication U.S. Patent No. 8,874,568, Octber 2014.

[7] H. Shah, M. Rezaei, and P. Fränti, "DOM-based keyword extraction from web pages," In procedings of Inteernational Conference on artificial intelligence, information processing and cloud computing (AIIPCC), Sanya, China, Article No. 62, 2019.

[8] 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.

[9] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma, "Extracting content structure for web pages based on visual representation," In proceedings of 5th Asia Pacific Web Conference, Xi'an China, 2003.

[10] 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.

[11] 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.

[12] S. Gupta, G. Kaiser, D. Neistadt, and P. Grimm, "Dom-based content extraction of html documents," In International Conference on World Wide Web ACM, New York, NY, USA, 2003.

[13] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma, "VIPS: a visionbased page segmentation algorithm," Microsoft technical report MSR-TR-2003-9, 2003.

[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] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-specific keyphrase extraction," Proceeding of IJCAI-99, pp. 668–673, 1999.

[16] O. Medelyan, and I. H. Witten, "Thesaurus based automatic keyphrase indexing," In ACM/IEEECS Joint Conference on Digital libraries, JCDL, pp. 296–297, 2006.

[17] C. Gutwin, G. W. Paynter, I. H. Witten, C. G. Nevill-Manning, and E. Frank, "Improving browsing in digital libraries with keyphrase indexes," In Journal of Decision Support Systems, pp. 81–104, 1999.

[18] Y. Xue, Y. Hu,G. Xin,R. Song, S. Shi, Y. Cao,C. Lin and H. Li, "Web page title extraction and its application," In Journal of Information Processing and Management, pp. 1332–1347, vol. 4, issue 5, 2007.

[19] 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, pp. 33–40, vol. 18, 2003.

[20] R. Mihalcea, and P. Tarau, "TextRank: Bringing order into texts," In proceeding of Conference on Empirical Methods in Natural Language Processing EMNLP, Barcelona, Spain, pp. 404–411, July 2004.

[21] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C .Nunes, and A Jatowt, "YAKE! Keyword extraction from single documents using multiple local features," In proceddings of Information Sciences, 509, pp. 257–289, 2020.

[22] M. Grootendorst, "KeyBERT: minimal keyword extraction with BERT," https://github.com/MaartenGr/KeyBERT, 2020.

[23] N. Gali, and P. Fränti, "Content–based title extraction from web page," In International Conference on Web Information Systems and Technologies (WEBIST), vol. 2, pp. 204–210, 2016.

[24] H. Shah, M. U. S Khan, and P. Fränti, "Hrank: a keywords extractionmethod from webpages using POS tags," In IEEE International Conference on Industrial Informatics (INDIN), Helsinki, 2019.

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

[26] M. Litvak, M. Last, H. Aizenman, I. Gobits, and A. Kandel, "DegExt – a language–independent graph–based keyphrase extractor," In Advances Intelligent Web Mastering, Springer Berlin Heidelberg, vol. 86, pp. 121–130, 2011.

[27] P. D. Turney, "Learning to extract keyphrases from text. National Research Council, Institute for Information Technology," Technical Report erb–1057, 1999.

[28] 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.

[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] T. Cover, and P. Hart, "Nearest neibhor patern classification," In IEEE Transactions on Information Theory, vol. 13–1, pp. 21–27, 1967.

[31] J. R. Quinlan, "Programs for Machine Learning," In Morgan Kaufmann, Los Altos, California, 1993.

[32] 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.

[33] V. Vapnik, "The nature of statistic learning theory," In Springer, New York, 1995.

[34] T. K. Ho, "Random decision forests," In International Conference on Document Analysis and Recognition, Montreal, Quebec, Canada, pp. 278–282 vol.1, 1995.

[35] F Rosenblatt, "Principles of Neurodynamic," New York: Spartan Books, 1962.

[36] P. Fränti and R. Mariescu–Istodor, "Soft precision and recall. Manuscript," Software: http://cs.uef.fi/paikka/Radu/tools/SoftEval/.

[37] 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.