H-Rank: A keywords extraction method from web pages using POS tags

Himat Shah
School of Computing
University of Eastern Finland
Joensuu, Finland
himat@cs.uef.fi

Muhammad U. S. Khan
School of Computing
University of Eastern Finland
Joensuu, Finland
usman@cs.uef.fi

Pasi Fränti
School of Computing
University of Eastern Finland
Joensuu, Finland
franti@cs.uef.fi

***Abstract*—We present a new keywords extraction method that applies the semantic similarity among the frequent words on the web page along with the distribution of POS tags. We apply hierarchical clustering to cluster the semantically similar words that have more coverage of the content of the web page. Our method shows better performance than CL-Rank and other existing methodologies.**

Keywords—Agglomerative clustering, POS tags, Web pages

I. INTRODUCTION

We are surrounded by an enormous wealth of information in the form of online web content, such as documents, databases, multimedia files, and web pages [1]. Access to the relevant web content depends upon the association of the web content with the *keywords*. International Encyclopedia of Information and Library Science [2] defines a keyword as a *word that succinctly and accurately describes the subject, or an aspect of the subject, discussed in a document.* Therefore, the keywords on a web page provide a compact representation of the web content [3] which can be used in many applications, such as automatic clustering, industry informatics, classification, or summarization [4]. Recently, natural language processing (NLP) based applications like keyword extraction is becoming very popular in the industry[13].

Manual keywords generation is an infeasible task due to the continuous growth of web pages. In the manual assignment of keywords, a fixed taxonomy is used by the professional curators [4]. Often users fail to find relevant information due to the absence of the quality keywords [6] due to the fixed autonomy. Therefore, automatic keywords generation is preferred over manual extraction. Automatic keywords generation is broadly divided into two approaches: *keywords assignment* and *keywords extraction* [5]. In the keywords assignment methodologies, a controlled vocabulary of words is used, whereas keywords extraction find all possible relevant words in a document [3].

The methodologies for keywords extraction from web pages differ from the keywords extraction from normal text documents [4]. The difference occurs because the text over the page is scattered in a different *hypertext markup language* (HTML), *JavaScript* (JS), and *cascade style sheet* (CSS) tags. Moreover, advertisements, navigational menus, and other sections of the web page include a huge amount of scattered text on the web page. The scattered text makes it difficult to extract the relevant information from the web page [4].

The keywords extraction methodologies from the web pages involve the usage of a *document object model* (DOM) tree. In the text nodes from the DOM tree are extracted for keywords [4] and title extraction [6, 7] from the page. The DOM structure has shown significant usefulness in the aforementioned studies. However, the content on web pages is increasing at a tremendous rate due to the usage of dynamic web pages and HTML5. This increase results in slow parsing of HTML and building of the DOM tree structure. Moreover, CSS trend of assigning new tasks to different well-known tags [8] affects the usage of DOM-based methodologies. In this work, we keep the usage of DOM structure as minimum as possible. Our method can therefore work also on documents as well with a slight modification.

Apart from the structure of the web pages, keyword extraction also depends upon the distribution of the part-of-speech (POS) tags in the text of the web page. The nouns cover most of the important content of the web page. Therefore, the distribution of nouns is considered as an important criterion for selecting the keywords in [4].

In this paper, we study the importance of the distribution of semantically similar POS tags, such as nouns, adjectives, and verbs in the extraction of relevant keywords from the web page. We have compared our methodology with the study [4] on the usefulness of nouns in keywords extraction, a graph-based approach *TextRank* [9], and *Term Frequency* (TF). TF indicates how many times a word appears on a web page [10].

We use four different publicly available datasets for the analysis. The keywords in the datasets are already assigned to the web pages by humans. The results show that a combination of nouns, adjectives, and verbs provide better keywords as compared to nouns alone. Following is the list of the contribution of our work:

- A new keywords extraction method that requires a minimum knowledge of DOM structure.
- The proposed method outperforms CL-Rank, TextRank, and TF.

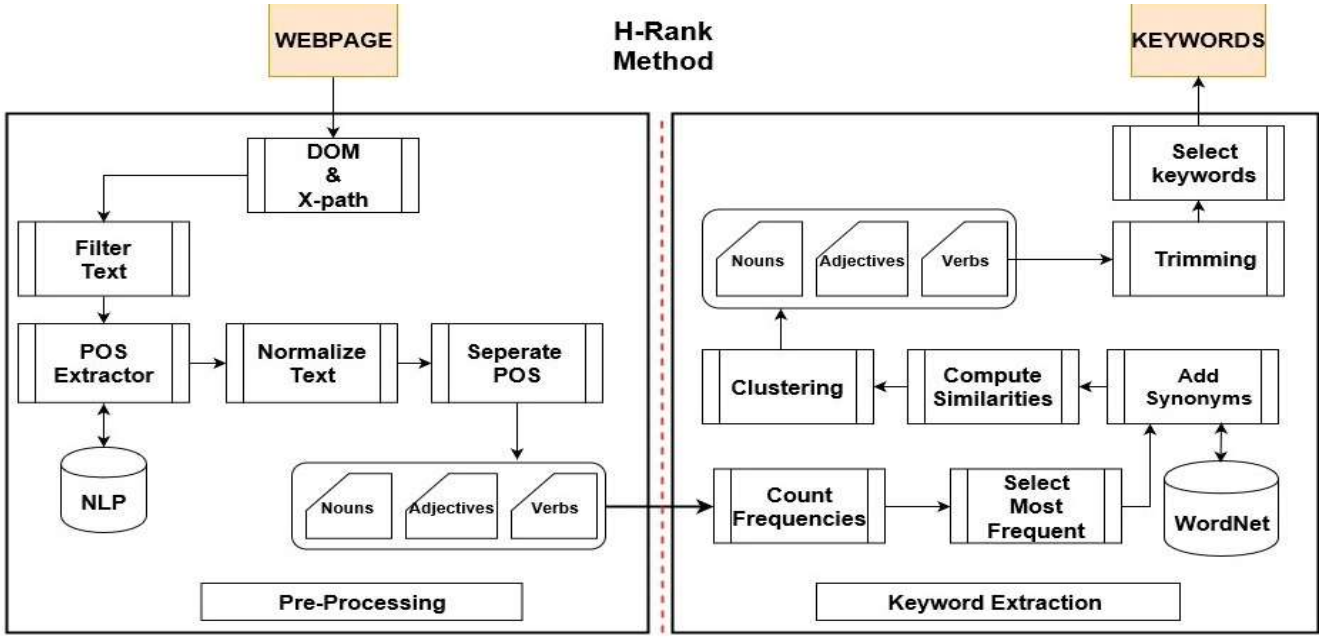

Fig. 1. Workflow of the proposed H-Rank

- A simple measure TF performs better than the more complex methods. However, the combination of nouns, adjectives, and verbs improves performance when TF fails.

The remainder of the paper is organized as follows. Existing keywords extraction methods are discussed in Section 2. The architecture of the proposed method is described in Section 3. Experimentation and results are discussed in Section 4 and Section 5 concludes the paper with a summary and a description of the future work.

II. KEYWORDS EXTRACTION

In this section, we will discuss some of the existing techniques in the field of keywords extraction The keywords extraction methodologies are grouped into four main categories: *statistical, linguistic, machine learning*, and *mixed* methodologies [3]. However, some authors categorize the keywords extraction methodologies into two categories: *corpus-oriented* and *document-oriented* [11].

The earlier works come under the corpus-oriented statistical category, such as [12] and [13]. In the corpus-oriented statistical methodologies, a list of statistically frequent words across a corpus is selected. The list is then utilized to select discriminating words as keywords from the individual document that separate a document from the rest of the documents. The techniques such as n-gram and TF measures are used for creating the list of the statistically frequent words.

The linguistic approaches are mostly domain-dependent and use lexical, syntactic, semantic, or discourse analysis [3]. The linguistic methodologies have shown better performance than the statistical methodologies [3, 14]. In machine learning approaches, a learning algorithm, such as *Naïve Bayes*, or a *decision tree* is used to classify the documents against a given set of keywords. However, the machine learning methodologies have a drawback of their dependence on the tagged training dataset [3].

The mixed methodologies involve any combination of statistical, linguistic, and machine learning algorithms. Recent *graph-based* keywords extraction methodologies come under this category. The graph-based methodologies are document-oriented or domain independent [3] and do not require training data. In [15] and [16], authors have used a graph-based approach mixed with statistical methods. In these methodologies, the words are represented as vertices and the edges are assigned values based on different statistical measures.

The keywords are extracted based on their edge values. The mixed-methodologies methods often use clustering techniques to group similar words together [15-17]. However, results in [4] showed that clustering-based techniques do not provide a significant improvement on web pages as compared to the statistical features, such as term-frequency. We have also found that humans tend to assign similar keywords or synonym keywords to the web pages and even term-frequency fails in those cases [4].

Our work falls into the mixed methodology category. We derive thresholds for adding the number of top-ranked adjectives and verbs with nouns in the keywords. To our surprise, the method performs better than the most similar existing approach. It also performs better than Text Rank a graph-based method for keywords extraction [9]. We have also compared our method with statistical TF method and it shows better results.

III. Proposed H-rank Method

Fig. 1. presents the workflow of the proposed keywords extraction method. The method has two modules: (1) pre-processing and (2) keyword extraction. The pre-processing module involves the extraction of the natural language text from the web page. The keyword extraction module utilizes the text from the pre-processing module.

In the pre-processing module, the first three functions involve the filtering of the text from all the other content of a web page. All the content of a web page is extracted using a *document object model* (*DOM*) and *X-path function*. The text that belongs to JavaScript scripting language and cascade style sheets is eliminated in the *text filtering* function. The special characters, such as @, *, £, or $, punctuation marks, and numbers are also filtered out using the regular expression in the text filtering function. Similarly, the text filtering function also involves the removal of the *stop words* from the text. The stop words are the natural language words that have minimal or no meaning, such as *and*, *the*, *a*, and *an* [18]. The filtered text can now be utilized for natural language processing.

The *POS extractor*, *normalize text*, and *separate POS* functions involve the natural language processing on the filtered text. The POS extractor function divides the text into tokens. A *token* is a whitespace-separated unit of text [19]. The tokens are assigned the POS tags, such as nouns, adjectives, and verbs.

The tokens with POS tags are further normalized. The normalization is the process of replacing the inflected forms of a word with the root word. The inflected form represents the different usage of a word in the sentences. For example, *finds, finding,* and *found* are the inflected forms of the word *find*. An inflected form of a word has a changed spelling or ending. In natural language processing, the lemmatization is used to find the inflected form of the words with different spellings, such as *finds* and *found* for the word *find* in the above example. Unlike lemmatization, the stemming process takes care of the prefixes and suffixes to find the root word, such as *finding* in the abovementioned example. The output of the normalization process is the tokens with all the inflected forms replaced with their root word.

The lists of the POS-tagged tokens are provided to the *separate POS* function, which separates the tokens into the lists of nouns, adjectives, and verbs. The lists are provided to the *count frequency* function. The count frequency function calculates the frequency of the words in the separate lists having nouns, adjectives, and verbs. The top-frequent tokens are selected as candidate keywords. The semantically similar words among top-frequent tokens are grouped together using a lexical database, named as WordNet. The lexical database helps in finding the synsets of the words. The *synset* is a set of one or more synonyms that can be used interchangeably in some context [20].

We compute the semantic similarity of two different words using path-similarity, which is based on the WordNet [21]. The words that have no synonyms in the WordNet are removed from the lists. The path-similarity metric calculates the score between two different words in terms of their relatedness. We use path-similarity because it is very simple and it operates based on a parent-child relationship like a tree. Therefore, it is more convenient to use in our case.

Three similarity matrices are created independently for the nouns, adjectives, and verbs. The similarity matrices are utilized in clustering the related words. We use an agglomerative clustering to find similar words in the lists [4]. The clusters are scored by counting the frequencies of all the words in each cluster. The clusters are ranked according to the scores.

The clusters with low scores are removed using the following equation:

$$ClusterScore < TS \cdot \max(ClusterScore) \qquad (1)$$

where TS is a trimming threshold and *max*(*ClusterScore*) is the score of the highest ranked cluster. Fig. 2 shows the precision, recall and F-measure scores at different trimming threshold values with nouns only. The best results are obtained at the value 0.30. The cluster scores provide the information that sometimes low-frequent words adds up together to form a cluster of a high score. Moreover, often the cluster does not show high score despite having one high frequent keyword. Therefore, the words that are individually frequent but do belong to a cluster of high score are also trimmed from the set of candidate keywords.

Table I shows the clusters from the list of nouns for an example webpage[22] from the Herald dataset. With equivalent to 0.3, only top 4 clusters are selected. The minimum score required to select a cluster is 12.6. The fifth cluster is dropped despite having a word *drivers* with high frequency. After removing the low-ranked clusters, the low-frequent words in individual clusters are removed using the trimming equation, similar to the equation presented in Ref. [4]. The equation is as follows:

$$TFrequency < 0.2 \cdot \max(TFrequency) \qquad (2)$$

where 0.2 is a trimming threshold and *maxFrequency* is the highest frequency of any word in the candidate keywords. The candidate keywords having a frequency lower than *TFrequency* are removed from the lists. The remaining clusters are merged together for each list. Fig 3. shows the lists of nouns, adjectives, and verbs with their frequencies after the trimming process.

We use the list of nouns as the main candidate keywords list. The top frequent words from adjectives list are added. Fig.4 shows the addition of different numbers of adjectives with the list of nouns. Addition of only a single adjective provides the highest precision and recall scores in our experiments. Similarly, top frequent verbs are added to the candidate keywords list after the addition of adjectives.

TABLE I. CLUSTERS FROM THE LIST OF NOUNS FOR A WEBPAGE

Cluster Rank	Elements and their frequencies	Cluster Score
1	Crash(17) Study(10) Type(5) Fusion(4) Engineering(3) Use (3)	42
2	University(13) Student(10) Gender(7) Loans(4) College(4) Percent(2)	40
3	Cell(4) Neutrons(4) Robots(4) Guide(4) Simple(4) Maps(4) Tips(4) Top(4) Herald(4) Kansas(3)	39
4	Severity(5) Injury(4) Males(4) Health(4) Safety(3) Age(3) Degree(2)	25
5	Drivers(10) Water(4)	14
6	Nov(4) Sep(3) May(2)	09
7	New(5)	05
8	Online(2)	02

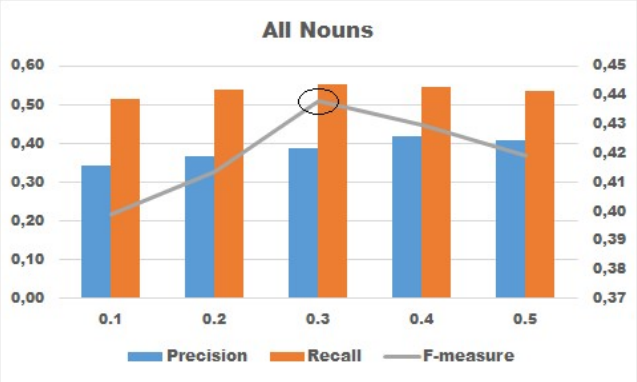

Fig. 2. Calculating trimming threshold using nouns from Herald news dataset

Nouns

Crash (17) University (13) Student (10) Study (10) Gender (7) Type (5) Severity (5) Cell (4) Neutrons (4) Males (4) Robots (4) Guide (4) Simple (4) Maps (4) Tips (4) Top (4) Herald (4) Fusion (4) Loans (4) College (4) Injury (4) Health (4)

Adjectives

Young (18) New (5) Simple (5) Direct (4) Membrane (4) Likely (4)

Verbs

Involved (7) Linked (5) Staying (4) Produce (4)

Fig. 3. Lists of nouns, adjectives, and verbs after trimming of frequency

The H-Rank keywords along with the ground truth, CL-Rank, Text Rank and TF method keywords are shown in Fig. 6. Unfortunately, the word driver that was removed in the cluster trimming appears in the ground truth for the toy example. However, five out of the top six frequent keywords in the H-rank keywords are available in the ground truth. The word student is also available in the ground truth with plural form but due to an exact match, it failed to be matched. Similarly, different forms of word crash also increased the complexity of finding all the words.

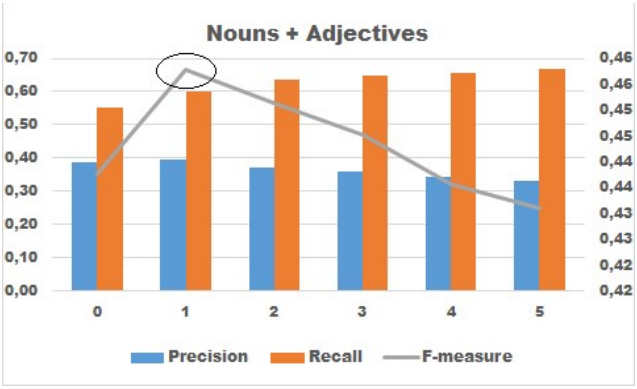

Fig. 4. Adding the different number of adjectives in the candidate keywords nouns from Herald news dataset

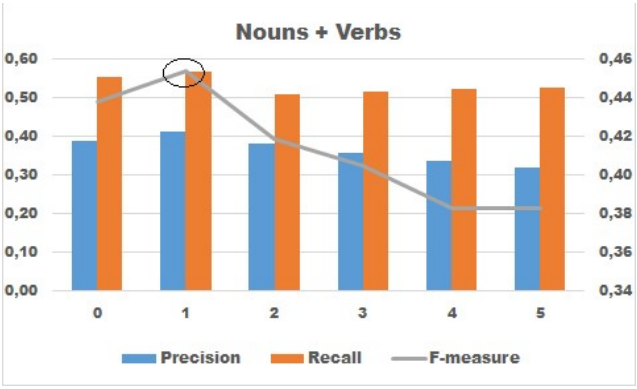

Fig. 5. Number of verbs added in the candidate keywords list on Herald dataset

Ground Truth

Drivers Study Students Gender Angeles Severity Los Game Males Crashes University Crash Guide

CL-Rank

Young Est Crashes Than Drivers University Involved Students Game Gender

Text Rank

Experts According to Pedestrians Statement Initiative Young University Crashes Drivers Gender

TF

Young University Crashes Drivers Gender Edt Study New Crash Tips

H-Rank

Crash University Study Student Gender Severity Type Cell Neutrons Males Robots Guide Simple Maps Tips Top Herald Fusion Loans College Injury Health Young Involved

Fig. 6. List of the keywords for different methods

IV. Experimentation and Results

We use four publicly available datasets[23]. The topics on the web pages are related to news, education, sports, health, politics, business, cities, entertainment, media, technology, and others. There are 500 webpages from NLM-500 and 421 web pages from the Guardian and 300 web pages from the University Herald and 100 webpages from Najlah dataset [4]. The reason for having four datasets and variety in categories and heterogeneity in web pages is to see how our method performs in general.

For the performance analysis, we use precision, recall, and F-measure scores to evaluate our methodology. Precision is the number proportion fraction of the correctly recognized keywords and measured as:

$$precision = \frac{Tp}{Tp + Fp} \tag{3}$$

A recall is the number proportion fraction of the keywords in the ground truth that are correctly recognized and can be measured as:

$$recall = \frac{Tp}{Tp + Fn} \tag{4}$$

F-measure is a classical accuracy measure and is a harmonic mean of precision and recall. It is calculated using the formula:

$$f - measure < \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{5}$$

Table II, Table III, Table IV, and Table V, present the precision, recall, and F-measure scores on the four different datasets. The average results of precision, recall and F-measure results are shown in Fig. 7 on combined all the four datasets. We have compared our method with CL-Rank [4] Text Rank [9] and a statistical measure TF.

TF was the simplest method in our study but it performed slightly better than the other methods in the NLM-500 dataset. However, due to the high F-score of the H-Rank, it outperformed TF and other methods in the F-measure. The results confirm our previous claim presented in [4] that only term-frequency is enough to find good quality keywords. However, removing the high-frequency words that do not participate in high score cluster and adding only a few highly frequent adjectives and verbs the average results improve and outperform all the other techniques.

The authors are of the view that overall low score of all the keywords extraction methods, especially the precision scores, are due to the limitation of the standardized evaluation methods. The ground truth assigned by humans is not constrained by any form or definition. On an exact match of words, a single word fails to be matched if a keywords extraction method finds a singular form whereas human used plural or vice versa. Similarly, it is also possible that human can use a synonym of a word that is picked up by the keywords extraction method that results in a low score despite being semantically found. Therefore, there is a need for a better method to evaluate the keywords extraction methods.

TABLE II. Performance Measurement on NLM-500 Dataset

Method	Precision	Recall	F-measure
CL-Rank	0.47	0.31	0.38
Text Rank	0.39	0.29	0.34
TF	0.45	0.33	0.39
H-Rank	0.48	0.37	**0.41**

TABLE III. Performance Measurement on Guardian Dataset

Method	Precision	Recall	F-measure
CL-Rank	0.17	0.21	0.18
Text Rank	0.12	0.16	0.14
TF	0.16	0.18	0.17
H-Rank	0.27	0.21	**0.23**

TABLE IV. Performance Measurement on Herald Dataset

Method	Precision	Recall	F-measure
CL-Rank	0.66	0.71	0.68
Text Rank	0.59	0.56	0.57
TF	0.63	0.60	0.61
H-Rank	0.78	0.90	**0.85**

TABLE V. Performance Measurement on Najlah Dataset

Method	Precision	Recall	F-measure
CL-Rank	0.49	0.48	**0.46**
Text Rank	0.33	0.37	0.35
TF	0.36	O.45	0.39
H-Rank	0.42	0.45	0.43

Fig. 7. Average results with the four datasets

V. CONCLUSIONS AND FUTURE WORKS

We present a new keywords extraction method that applies the semantic similarity among the frequent words on the web page. Moreover, we have applied hierarchical clustering to group semantically similar words that have more coverage of the content of the web page and removed the words that do not make up a high ranked cluster. We have found a good number of required adjectives and verbs in the keywords extraction method that can increase the F-measure scores. Our method has shown comparable performance to term-frequency CL-Rank and other existing methodologies. In the future, we plan to improve the keywords extraction method using Word2Vec gensim library replacing WordNet.

REFERENCES

[1] A. Díaz A. Rios J. Barron T. Guerrero and J. Elizondo "An automatic document classifier system based on genetic algorithm and taxonomy" IEEE Access 2018.

[2] J. Feather and P. Sturges International encyclopedia of information and library science: Routledge 2003.

[3] A. Onan S. Korukoğlu and H. Bulut "Ensemble of keyword extraction methods and classifiers in text classification" Expert Systems with Applications vol. 57 pp. 232-247 2016.

[4] M. Rezaei N. Gali and P. Fränti "CL-Rank: A method for keyword extraction from web pages using clustering and distribution of nouns" in Web Intelligence and Intelligent Agent Technology (WI-IAT) 2015 IEEE/WIC/ACM International Conference on 2015 pp. 79-84.

[5] S. Siddiqi and A. Sharan "Keyword and keyphrase extraction techniques: A literature review" International Journal of Computer Applications vol. 109 2015.

[6] N. Gali and P. Fränti "Content-based title extraction from web page" in WEBIST (2) 2016 pp. 204-210.

[7] N. Gali R. Mariescu-Istodor and P. Fränti "Using linguistic features to automatically extract web page title" Expert Systems with Applications vol. 79 pp. 296-312 2017.

[8] A. S. Bozkir and E. A. Sezer "Layout-based computation of web page similarity ranks" International Journal of Human-Computer Studies vol. 110 pp. 95-114 2018.

[9] R. Mihalcea and P. Tarau "Textrank: Bringing order into text" in Proceedings of the 2004 conference on Empirical methods in Natural Language Processing 2004.

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

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

[12] K. Sparck Jones "A statistical interpretation of term specificity and its application in retrieval" Journal of Documentation vol. 28 pp. 11-21 1972.

[13] 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.

[14] A. Hulth "Improved automatic keyword extraction has given more linguistic knowledge" in Proceedings of the 2003 conference on Empirical methods in Natural Language Processing 2003 pp. 216-223.

[15] G. K. Palshikar "Keyword extraction from a single document using centrality measures" in International Conference on Pattern Recognition and Machine Intelligence 2007 pp. 503-510.

[16] 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.

[17] D. B. Bracewell F. Ren and S. Kuriowa "Multilingual single document keyword extraction for information retrieval" in Natural Language Processing and Knowledge Engineering 2005. IEEE NLP-KE 05. Proceedings of 2005 IEEE International Conference on 2005 pp. 517-522.

[18] W. J. Wilbur and K. Sirotkin "The automatic identification of stop words" Journal of Information Science vol. 18 pp. 45-55 1992.

[19] B. Medlock "An introduction to nlp-based textual anonymization" in Proceedings of 5th International Conference on Language Resources and Evaluation (LREC) Genes Italie 2006.

[20] G. A. Miller and C. Fellbaum "Wordnet then and now" Language Resources and Evaluation vol. 41 pp. 209-214 2007.

[21] 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.

[22] http://www.universityherald.com/articles/11102/20140827/gender-crash-injury-severity-kansas-young-driver.htm
http://cs.uef.fi/mopsi/keywords/

[23] https://code.google.com/archive/p/maui-indexer/downloads