Transactions on Machine Intelligence 5(2) (2022) 122–138

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

# Implementation of a System for Removing Noisy Hyperlinks: A Semantic and Relatedness-Based Approach

K. Taghandiki[1,*], E. Rezaei Ehsan[2]

[1] Department of Computer Engineering, Technical and Vocational University (TVU), Tehran, Iran
[2] Master's Degree, Industrial Engineering, System Management and Productivity, Iran University of Science and Technology, Tehran, Iran

**ARTICLE INFO**

Article History:
Received 2 April 2022
Received in revised form
23 May 2022
Accepted 29 June 2022
Available online 30 June 2022

Keywords:
Semantic Web, Noisy Hyperlinks, Ontology, Reasoner, Semantic Similarity, Relatedness Similarity

**ABSTRACT**

With the continuous growth of information on the web, the complexity of the web structure graph an abstract representation of the web as a network of interconnected nodes has also evolved significantly. Traditionally, these structures were content-based, but recent developments have led to a shift toward non-content-based architectures. One of the major challenges in this evolving landscape is the proliferation of spam data, particularly noisy or irrelevant hyperlinks. These unwanted links can degrade the performance of information retrieval systems and link mining algorithms by introducing redundancy and misleading connections. Previous approaches have primarily relied on structural or string similarity techniques to filter out noisy hyperlinks. However, such methods often suffer from limitations, including the potential removal of semantically important links or the failure to detect subtle noise. To address these challenges, this study introduces a semantic web-driven framework. We first build a dataset of hyperlinks using an interactive web crawler. Next, we assess the semantic relevance and interlinkage of these hyperlinks using tools such as the DBpedia ontology. The process of noisy hyperlink elimination is carried out using a reasoning engine applied over the ontology. Experimental results confirm that semantic web technologies offer improved precision and robustness in identifying and eliminating noisy hyperlinks, thereby enhancing web data quality and usability.

## 1. INTRODUCTION

The ubiquity of hyperlinks in digital content has revolutionized information access, but the prevalence of noisy hyperlinks poses a formidable challenge to the quality of user experience and content relevance. This paper delves into the design and implementation of a sophisticated system aimed at mitigating the impact of noisy hyperlinks through a novel Semantic and Relatedness-Based Approach. By leveraging cutting-edge natural language processing

* Corresponding Author: taghandiky@gmail.com
Department of Computer Engineering, Technical and Vocational University (TVU), Tehran, Iran



http://dx.doi.org/10.47176/TMI.2022.122

and semantic analysis techniques, the system aspires to enhance the precision and contextuality of hyperlinks in digital content [1-3].

In recent years, the concept of 'Big Data' has emerged due to the ability to generate vast amounts of data in various technological fields. According to Google and Bing crawlers, nearly 49 billion web pages were indexed in 2021 [4], indicating a significant increase in the number of web pages on the Internet and the growth of the web structure graph. Navigating and exploring the structure of the web can be challenging due to spam data, such as noisy hyperlinks [5]. Therefore, a mechanism is necessary to eliminate spam hyperlinks. Historically, information retrieval algorithms have used the contents of web documents to classify, cluster, and remove spam pages. However, the process of relying on document content, which was once time-consuming and process-intensive, has been replaced by algorithms that utilize hyperlink characteristics in the web structure graph. Several algorithms, such as PageRank [6], use these characteristics to reduce the processing required for search engines. However, these link-based algorithms assume that the links point precisely to the pages desired by the users [7]. Link mining algorithms often assume that the web structure graph is entirely semantic and content-based, but this is a mistake. The graph contains spam links that can mislead users and affect the algorithm's output. Spam hyperlinks allow irrelevant documents to obtain higher ranks than relevant ones, and this attempt to boost a page's rank is mainly for business purposes [7]. Several studies have been conducted to detect and eliminate spam links from web structures. However, the proposed methods heavily rely on hyperlink strings and structural characteristics [7,8], while ignoring their semantic and relatedness structures.

This paper examines the semantic and relatedness structures of hyperlinks at both the page and site levels. Semantic web technologies, such as ontologies and reasoners, are used to eliminate noisy hyperlinks.

A dataset of hyperlinks is first created in a separate process. Then, the concepts of the source page hyperlinks and the target page are semantically and relationally analyzed using ontologies and reasoners. The analysis can determine whether a hyperlink is useful or noisy.

The proposed system takes the constructed dataset as input. Each row of the dataset includes the class mapped from the hyperlink context topic of the source page, the class mapped from the topic of the target page, a field indicating the noisy or useful nature of the hyperlink from the user's perspective, and the domain name of the source page. Afterwards, the system detects noisy hyperlinks and compares the results to those of the user to determine the contribution of each semantic or relatedness property in the ontology to correct detection. Additionally, this approach allows for identification of the queries that lead to noisy hyperlinks and the domains with the highest number of them. The experiments demonstrate the accuracy, capability, and scalability of semantic web technologies in eliminating noisy hyperlinks.

The paper is organized as follows: Section 2 provides a survey of previous works on the removal of noisy hyperlinks, Section 3 details the implementation of the proposed approach, Section 4 explains the experiments and the obtained results, and Section 5 presents concluding remarks.

## 2. RELATED WORK

Qi et al. [9] classify hyperlinks as navigational, advertising, irrelevant, or useful. They propose an algorithm that uses a Support Vector Machine (SVM) with two classes, 'qualified' and 'unqualified', to detect and filter noisy hyperlinks. The algorithm employs six string similarity features and is applied to a collection of 2.1 million web pages. The results show that 23% of the hyperlinks are classified as 'unqualified'. However, the algorithm does not use semantic or relatedness approaches to remove hyperlinks.

Wookey et al. [10] design a system called Anchor Woman, wherein noisy hyperlinks are detected using the hyperlink structure of a website and divided into three categories:

- Multi-arc loops: chains of hyperlinks which form many cycles in the web graph.
- Multiple arcs: many hyperlinks that point to the same page.
- Recursive cycles: web pages that contain hyperlinks pointing to themselves.

The system operates by taking a web address as input and performing a breadth-first search of its hyperlinks to identify and remove noisy ones. Then, it generates a graphical hierarchical representation of the web structure for the user. Carvalho et al. [11] propose a mechanism for detecting noisy hyperlinks at the site level. They identify two types of spam relationships: (1) mutual reinforcement, where two websites are strongly connected by exchanging site-level hyperlinks, and (2) alliance among a chain of strongly connected websites. If the number of hyperlinks between the sites exceeds a threshold, they are considered noisy. The algorithm proposed takes a structural approach and is capable of removing 16.7% of hyperlinks with a Mean Average Precision of 59.16%.

Chakrabarti [12] proposes a more detailed model of the web, in which pages are represented by their Document Object Models. The resulting DOM trees are interconnected by regular hyperlinks. This method can counter 'nepotistic clique attacks,' but requires more input data than our algorithms, which are based solely on link analysis. Also, as we focus on noise removal, we can identify various types of hyperlink anomalies.

Samanta et al. [8] use graph-based methods to enhance the web structure graph and facilitate user navigation. The study examines UK university websites, extracting over six million links from 110 academic websites to form a dataset. A significant number of undesirable links to images, audio, and video files are eliminated using TextPipe. This approach optimizes the number of web documents, path length, and Strongly Connected Components (SCC).

However, the proposed methodology relies merely on the type of hyperlink, without considering semantic or relational approaches. The two steps are as follows:

1. Eliminating advertising and navigational hyperlinks which are commonly located near the top of the page.
2. Eliminating the hyperlinks not covered by association or aggregation relationships.

An aggregation relationship is a hierarchy relationship between two concepts where the source concept is broader than the target concept. An association relationship, also known as a horizontal relationship, implies that the source and target concepts share the same parent. In other words, two concepts are horizontally related if and only if they have a common parent. The authors reported a recognition rate of 92.89 percent in removing navigational hyperlinks. The aggregation relationship conveys a semantic approach, while the association relationship cannot be considered a complete approach to relatedness. Pedersen et al. [13] use an association or horizontal relationship to demonstrate relatedness similarity between two concepts, but horizontal relationships only indicate a 'Part Of' relationship between the two. However, other relational properties, such as object properties in ontologies, can also represent similarity in relatedness.

Oguz [14] proposed another algorithm for removing noisy hyperlinks in 2022, called the Website Structuring Extracting Algorithm (WSE). The primary goal of the WSE is to eliminate noisy hyperlinks while retaining semantic ones. However, the paper focuses on the path structure of hyperlinks to maintain the hierarchy of hyperlinks. For instance, assuming four pages (A, B, C, and D), hyperlinks from A to B, B to C, and C to D are semantic hyperlinks, while hyperlinks in the opposite direction are considered noisy.

In [15], the authors propose a method for detecting nepotistic links using language models. The method down-weights a link if its source and target pages are not related based on their language models. This approach assumes that pages connected by non-nepotistic links must be sufficiently similar.

Wu and Davison [16] propose a two-step algorithm for identifying link farms. The first step generates a seed set based on the intersection of in-links and out-links of web pages. The second step involves expanding the seed set to include pages that link to many pages within the seed set. The links between these identified spam pages are then re-weighted, and a ranking algorithm is applied to the modified link graph.

Previous works have tended to focus on page-level hyperlinks. However, modern spam sites usually make use of site-level hyperlinks by generating illegal links to other websites, thereby improving their rank in Google's index. Therefore, it is important to consider hyperlinks at the site level. This paper aims to remove noisy hyperlinks at both the page and site levels. Previous studies have been limited by their exclusive use of string or structural approaches. While these approaches are fast at detecting hyperlink types, they may eliminate useful hyperlinks and fail to detect noisy ones. For example, consider a scenario where a page with the term 'Bank' (referring to a financial institution)

links to a page about 'Banks' (referring to a shore). The string approach would consider this hyperlink useful, while the semantic approach would use the hyperlink text information to identify and remove irrelevant hyperlinks. This paper applies the semantic web approach and current tools, such as the DBpedia ontology, to analyze the semantic and relational structure of hyperlinks and remove noisy hyperlinks by activating the DBPedia ontology reasoner.

The experiments demonstrate the accuracy and effectiveness of semantic web technologies in eliminating noisy hyperlinks. In contrast to preceding mechanisms, such as [5,17,14,18], which do not use existing data collections for information retrieval, the web structure graph is analyzed to remove noisy hyperlinks. Therefore, it is necessary to construct a data collection of hyperlinks. The next section presents the details of this procedure based on information retrieval principles [19].

## 3. PROPOSED APPROCH

The semantic and relatedness system for eliminating noisy hyperlinks involves three general steps as shown in Figure 1.

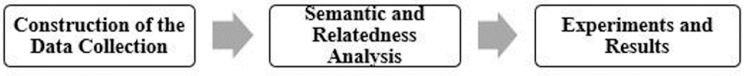


**Fig. 1.** Implementation process of the proposed approach

### 3.1. Constructing the Data Collection

The dataset construction step is a distinct process consisting of several steps as shown in Figure 2.

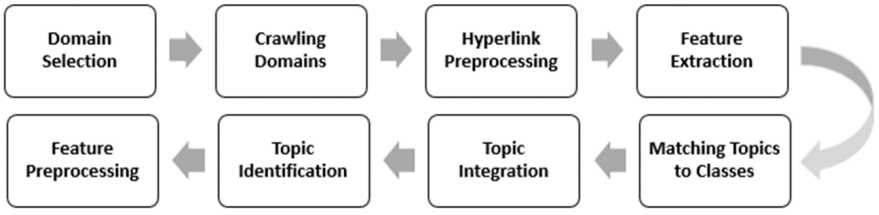


**Fig. 2.** Stages of constructing the data collection

In constructing the dataset, the user is only involved in domain selection while the other steps are performed independently.

#### *3.1.1. Domain Selection*

Every day, millions of internet users submit queries to search engines like Google for various purposes. The selection of websites to crawl is a crucial issue, as both useful and noisy hyperlinks are necessary. The proposed approach demonstrates its ability to detect useful hyperlinks using semantic and relatedness properties of the ontology, while eliminating the latter. The crawled websites must contain content that is popular among Internet users. To identify popular topics, Google Trends was used, which revealed news, money, online shopping, new technologies, and celebrities such as actors or athletes. Table 1 displays popular search queries in 2021 according to Google Trends.

**Table 1.** Popular search queries in 2021

| Electronics | Sports | Celebrities | News |
|---|---|---|---|
| iPhone 13 | Real Madrid CF | Jenifer Lawrence | Afghanistan |
| Galaxy Z Flip4 | Chelsea F.C | Kim Kardashian | AMC Stock |
| Nexus Summit | Paris Saint-Germain F.C. | Julie Gayet | COVID Vaccine |
| Motorola Moto G Power | FC Barcelona | Tracy Morgan | Dogecoin |

By learning about popular topics among Internet users, organizations and individuals can take two distinct approaches in designing web pages.

1. Creating websites that are weakly focused on the topic but use background hyperlinks to conduct highly profitable business activities such as directing users to online stores or pornography websites. Such websites contain noisy hyperlinks which have no regard for web user needs.
2. Creating websites with useful content on a particular topic to provide users with appropriate information. Hyperlinks in these websites are rarely considered spam. These websites contain useful hyperlinks that are in line with user needs.

The main idea behind this approach to domain selection is to crawl domains retrieved by search engines in response to frequent queries. The retrieved domains fall into one of two general categories: (1) those with noisy hyperlinks for illegal business purposes and (2) those with legal objectives that provide useful hyperlinks to help users achieve their goals. The proposed method's strength in maintaining useful hyperlinks while removing noisy ones is demonstrated by its ability to distinguish between these categories.

### *3.1.2. Crawling Domains*

During this stage, the user inputs each topic from the previous stage into the Google search engine and selects a random subset of the returned websites. The website addresses are then inputted into an interactive crawler developed using Java programming language libraries. The crawler begins exploring the domain and obtains a list of links in the domain, as shown in Figure 3.


**Fig. 3.** Links extracted by the crawler

A total of 114 domains related to the topic were crawled, both useful and useless. All domains are in English.

### *3.1.3. Hyperlink Preprocessing*

Many of the extracted hyperlinks, such as repetitive links or those pointing to audio or video files, are not relevant to the purpose of this paper. Therefore, they were removed using the operations proposed by [5] (Section 2). The status of the hyperlinks after preprocessing can be seen in Figure 4.

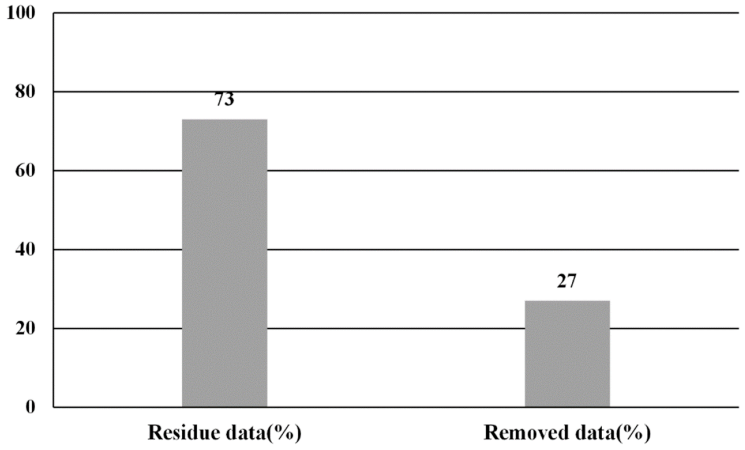


**Fig. 4.** Status of the hyperlinks subsequent to preprocessing

After preprocessing, 27% of the hyperlinks were removed, reducing the number of crawled hyperlinks from 2665 to 1946. It is important to note that many of the extracted domains contained video, image, and audio hyperlinks, which are incompatible with the proposed approach. Additionally, some links appeared multiple times on a page, resulting in the removal of 719 links, or 27% of the total. As previously stated, the proposed method is only compatible with text hyperlinks. Therefore, the subsequent steps, which concern dataset construction, are only applicable to text hyperlinks. Incompatible hyperlinks must be removed during the preprocessing step.

### *3.1.4. Feature Extraction*

To ensure accurate detection of the topic of a hyperlink's surrounding text and target page, it is necessary to extract several features from web pages. Identifying the most commonly used features in web design is crucial in determining which features to extract. Figure 5 presents the frequencies of the five key features in 5000 pages. As demonstrated, web pages most commonly utilize 'Keyword Metadata', 'Title Tag', and 'First-Level Heading Tag'.

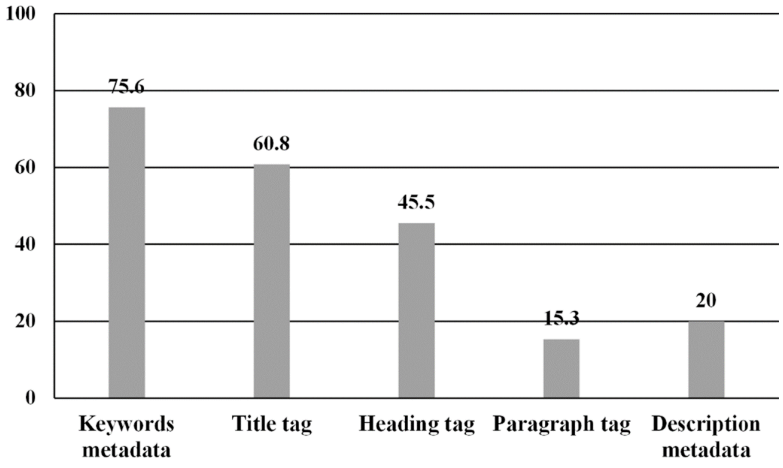


**Fig. 5.** Top five most commonly used features in web pages

Therefore, the topic of the target page is determined by analyzing three features: the title tag, keyword metadata, and first-level heading tag. Additionally, the text of the hyperlink and the paragraph containing it are used to extract the topic context of the hyperlink from the source page. The hyperlink text and its surrounding paragraph serve as

features that provide context for the hyperlink. Tables 2 and 3 provide examples of extracted features used to determine the topic of the target page and the context of hyperlink text, respectively.

**Table 2.** Extracted features to detect target page topic (from www.Filehippo.ocm)

| Page Title | Download free Software |
|---|---|
| **Keyword Metadata** | download software freeware shareware program |
| **First-level Heading** | Software |

**Table 3.** Extracted features to detect hyperlink text topic context (from www.Filehippo.com)

| Page Title | Download free Software |
|---|---|
| **Keyword Metadata** | download software freeware shareware program |
| **Hyperlinktext** | New Software |
| **Hyperlink paragraph** | The Latest Versions of the New Software |

These features speed up the analysis and topic detection procedures in subsequent stages. In contrast to our work, several studies [11, 20-22] extract topics based on the entire content of documents.

Most web designers use Web 2.0 techniques, such as HTML, to create web documents. However, this markup language is less semantically capable compared to its Web 3.0 counterparts, such as RDF, XML, and OWL. Thus, due to the popularity of HTML attributes in designing web documents, it seems that the language is the best choice for feature selection in Web 2.0. However, Web 3.0 techniques make it easy to semantically extract features, which is important in the proposed approach. Nonetheless, this paper does not cover this topic, and it is recommended for future works.

### *3.1.5. Feature Preprocessing*

The quality of the features extracted in the previous step must be improved so that the semantic and relatedness system is able to perform the topic identification process with higher accuracy and lower error rate. In this paper, this is achieved by using typical text mining preprocessing techniques such as stop words, token normalization, case folding, and stemming [19]:

1. Stop words refer to words which occur frequently and are of little use in finding specific information. Examples include articles and prepositions such as “the”, “and”, “or”, etc. Removing these words accelerates the topic detection step.
2. Token normalization is a standardization process which aims to use a single form for each word. For instance, consider “anti discriminatory” and “non discriminatory”; subsequent to the normalization process, both forms are mapped onto “anti-discriminatory”.
3. Case folding is a well-known word normalization process wherein capitalized letters are converted to their lower-case equivalents. In fact, case folding may be regarded as a type of token normalization.
4. Words are often used in different forms depending on grammatical rules; for example, organize, organizes, and organizing. Furthermore, different forms of a word may have nearly similar meanings e.g. democracy and democratic. By removing the endings of the words, the Stemming process aims to obtain a common root for different forms of a word.

In this study, preprocessing is performed via MALLET as well as the text mining library in Python.

### *3.1.6. Topic Identification*

These features speed up the analysis and topic detection procedures in subsequent stages. In contrast to our work, several studies [11, 20-22] extract topics based on the entire content of documents.

The aim of this stage is to identify the topic of hyperlink text and the corresponding target page based on the extracted features. This stage involves a supervised process with three operations, as shown in Figure 6, all of which are carried out using MALLET. The following subsections provide relevant details.

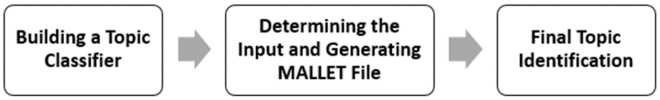


**Fig. 6.** Supervised topic detection process

### *3.1.7. Building a Topic Classifier*

To construct a topic classifier, it is necessary to have an initial set of high-quality features. Therefore, we extracted the three most common key features (i.e. keyword metadata, title tag, and first-level heading tag) from 5000 web pages to create a dataset of features. The topic classifier was built using this dataset. Additionally, 80% of the data was used for training, while the remaining 20% was used to test the accuracy of the classifier.

This paper conducts topic classification using four different methods: Naïve Bayes, C4.5, Decision Tree, and Max Entropy with 10 cross-validations [23], to achieve high efficiency. The algorithms are compared to determine the best method for classification. As shown in Figure 7, Max Entropy yields the highest accuracy on the training data, and is therefore used to create the topic classifier.

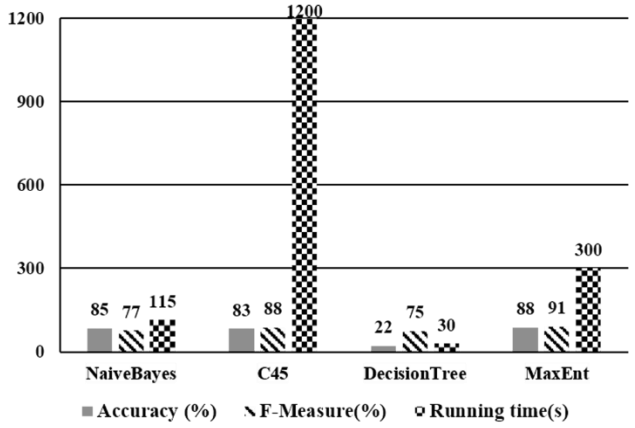


**Fig. 7.** Performance comparison of Naïve Bayes, Decision Tree, C4.5, and Max Entropy in terms of accuracy and execution time on the features dataset

### *3.1.8. Determining the Input and Generating MALLET File*

The task involves extracting features from hyperlink text context and target page and inputting them into MALLET to identify their respective topics. This is done through the command-line interface, resulting in two output files, also known as feature vectors. These vectors are numerical representations of the input values that enable faster analysis operations [23], and serve as input for the next operation.

### *3.1.9. Final Topic Identification*

Here, the MALLET output files from the previous operation are obtained and the topic classifier identifies the topic. Moreover, the ensuing model (i.e. output) can be used to infer the topic of new input data in the subsequent steps.

Upon completion, the topic classifier is able to successfully identify the topic of approximately 81.8 percent of extracted features. The remaining 18.2 percent of the features may remain unclassified for one of the following reasons:

1. As a result of poor design, the feature extraction step may be unable to extract appropriate features for identifying the topic of the hyperlink text and the target page. This situation precludes topic assignment.
2. The features used to construct the topic classifier may be completely distinct from those extracted from a new page. Consequently, the page is not assigned a topic.

The output of the step includes four separate text files. Table 4 presents a portion of each file.

**Table 4.** Sample output obtained after the final topic identification step

| File 1 | File 2 | File 3 | File 4 |
|---|---|---|---|
| Bird | Fish | 0 | **Nationalgeographic** |
| Bird | Mammal | 0 | **Nationalgeographic** |
| Car | Canvas | 1 | **Dairyfoods** |
| Song | Movie | 1 | **Songsmp3** |
| Music | Game | 1 | **Songsmp3** |

### *3.1.10. Topic Integration*

The topics from the previous stage, located in several text files, are combined and integrated to create a single well-formed input for the matching stage. An example of the output file is shown in Figure 8.

```
Bird,Fish,0,nationalgeographic
Bird,Mammal,0,nationalgeographic
Car,Canvas,1,dairyfoods
Song,Movie,1,songsmp3
Music,Game,1,songsmp3
Wine,Food,0,dairyfoods
```

**Fig. 8.** Topic integration output file

The file's fields, from left to right, represent the context topic of the hyperlink in the source page, the topic of the target page, the type of link (i.e. noisy or useful) as perceived by the user, and the domain of the source page. This stage constructs a topic dataset.

### *3.1.11. Matching Topics to Classes*

The aim of this stage is to align the topics from the previous stage with the classes of the DBpedia ontology. This is achieved through the use of the WS4J library and the Terminological Search Algorithm (Semantic Search). Figure 9 presents a comparison of the two algorithms using 400 randomly selected topics and five criteria, including accuracy.

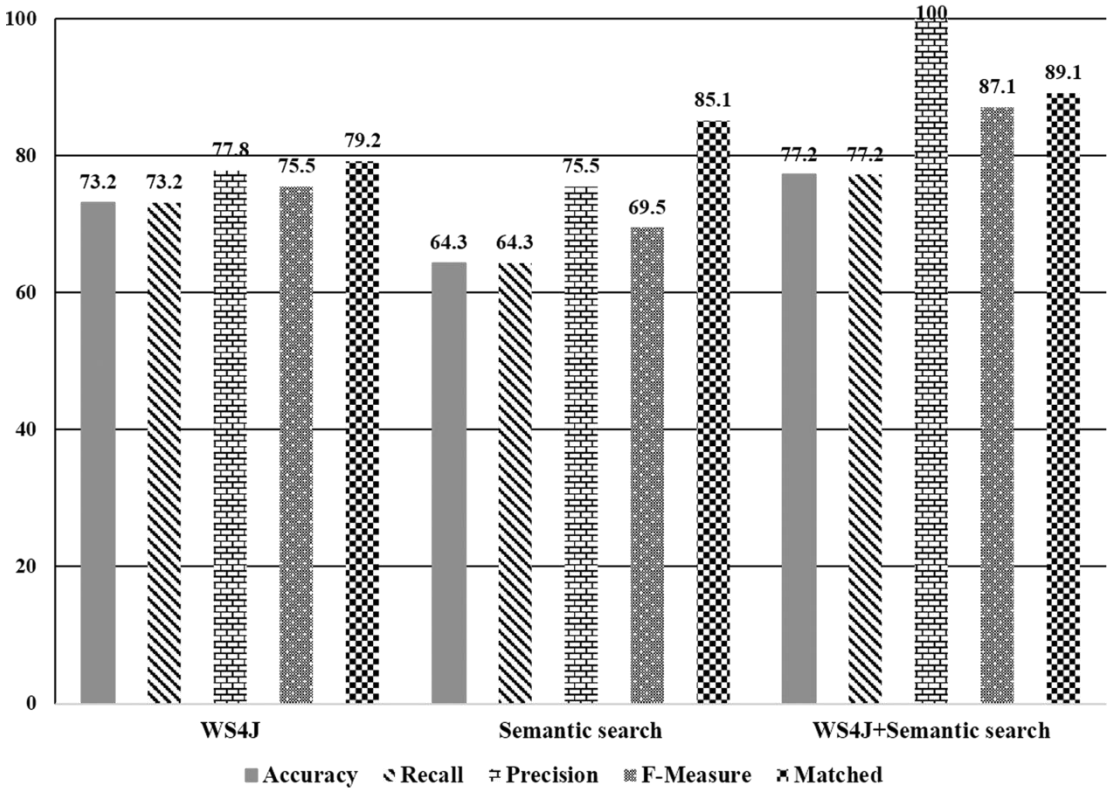


**Fig. 9.** Comparison of WS4J, Semantic Search, and WS4J+Semantic Search

Based on the results, it is justifiable to examine the simultaneous application of WS4J and Semantic Search as they complement each other. Based on the results, it is justifiable to examine the simultaneous application of WS4J and Semantic Search as they complement each other. Figure 9 shows that the combination of both algorithms outperforms each individual algorithm. Additionally, both algorithms support the required semantic and string similarity, making WS4J+Semantic Search the preferred choice for matching purposes. Examples of the final matches can be found in Figure 10.

| | | | |
|---|---|---|---|
| http://dbpedia.org/ontology/Automobile | http://dbpedia.org/ontology/Automobile | 0 | ebay |
| http://dbpedia.org/ontology/Currency | http://dbpedia.org/ontology/Actor | 1 | mydailyfundose |
| http://dbpedia.org/ontology/Continent | http://dbpedia.org/ontology/Actor | 1 | mydailyfundose |
| http://dbpedia.org/ontology/Activity | http://dbpedia.org/ontology/Actor | 1 | mydailyfundose |
| http://dbpedia.org/ontology/Coach | http://dbpedia.org/ontology/Actor | 1 | mydailyfundose |
| http://dbpedia.org/ontology/MusicFestival | http://dbpedia.org/ontology/Actor | 1 | mydailyfundose |
| http://dbpedia.org/ontology/Economist | http://dbpedia.org/ontology/Actor | 1 | mydailyfundose |
| http://dbpedia.org/ontology/Food | http://dbpedia.org/ontology/Actor | 1 | mydailyfundose |
| http://dbpedia.org/ontology/NaturalPlace | http://dbpedia.org/ontology/Actor | 1 | mydailyfundose |

**Fig. 10.** Examples of matches

Once again, the fields represent the class matched to the topic context of the hyperlink in the source page, the topic of the matched class in the target page, the type of the hyperlink, and the domain name, respectively. Using the mapping subsystem, nearly 87.3 percent of topics are mapped to those of the DBpedia ontology. This step creates a final dataset referred to as the conceptual hyperlink dataset.

### 3.2. Semantic and Relatedness Analysis

At this stage, the system receives the final data collection as input, which includes two ontology classes: the context of the hyperlink text and the target page. The system then analyzes the relatedness and semantic properties of the input using the knowledge from the DBpedia ontology. This analysis helps to determine whether the hyperlink is useful or noisy.

The reasoner plays a critical role in the semantic and relatedness analysis step. The software operates on one or more conceptual datasets created using ontologies. Its purpose is to extract logical results from existing facts in the ontology.

The Pellet reasoner is used to achieve this task by utilizing primary knowledge from the DBpedia ontology and obtaining additional knowledge from the ontology's properties, relations, and classes. The following row of data collection presents a hyperlink in a web page, consisting of two concepts: hyperlink text context and target page. To determine the semantic and relatedness similarities between the two, we use properties and relations inferred from the DBpedia ontology. Throughout the rest of this paper, we will refer to the concepts of hyperlink text and target page as source concept and target concept, respectively. A hyperlink is considered useful, by the reasoner, if at least one of the following properties is satisfied:

1. “Equivalent Class”: The source and target concepts are equivalent.
2. “Subclass Of” and “Has Superclass”: The source concept is a sub/superclass of the target concept. Properties (1) and (2) are known as semantic properties, which represent semantic similarity between the two concepts. For instance, the concepts of “Woman” and “Person” are semantically similar since the former is a subclass of the latter.
3. “Object Property”: The source and target concepts are related through an object property. This is known as a relatedness property and represents relatedness similarity. As an example, the concepts of “Monkey” and “Banana” have relatedness similarity through several relations such as “Liking” or “Eating” (i.e. “The monkey likes bananas” or “The money eats bananas”).

Many current methods for removing noisy hyperlinks only consider the first two characteristics and ignore Objectivity. This can lead to the removal of useful hyperlinks because they fail to detect relatedness properties between the source and target concepts.

If the source and target concepts have no semantic or relatedness similarity, the hyperlink is considered noisy. Put simply, instead of directing the user to their intended page, the hyperlink on the page leads to an unexpected and irrelevant page. Therefore, at the end of this step, a conceptual dataset of hyperlinks is created, and its usefulness is determined by the reasoner. Figure 11 displays several records constructed during the analysis stage.

| | | | | |
|---|---|---|---|---|
| http://dbpedia.org/ontology/Automobile | http://dbpedia.org/ontology/Automobile | subClassOf | 0 | ebay |
| http://dbpedia.org/ontology/Cartoon | http://dbpedia.org/ontology/Agent | animator | 0 | wikipedia |
| http://dbpedia.org/ontology/Person | http://dbpedia.org/ontology/Place | birthPlace | 0 | wikipedia |
| http://dbpedia.org/ontology/Person | http://dbpedia.org/ontology/Place | livingPlace | 0 | wikipedia |
| http://dbpedia.org/ontology/Film | http://schema.org/Movie | equivalentClass | 0 | athlete |
| http://dbpedia.org/ontology/MusicFestival | http://dbpedia.org/ontology/Actor | | 1 | mydailyfundose |
| http://dbpedia.org/ontology/Economist | http://dbpedia.org/ontology/Actor | | 1 | mydailyfundose |
| http://dbpedia.org/ontology/Food | http://dbpedia.org/ontology/Actor | | 1 | mydailyfundose |

**Fig. 11.** Records constructed during the analysis stage

The fields are arranged from left to right as follows: Subject, Object, inferred relational and semantic property for the relation between the first two fields, type of hyperlink as perceived by the reasoner, and the domain of the source page.

### 3.3. Experiment and Results

A confusion matrix is an important and useful tool for evaluating the proposed approach. It involves two types of labeling: (1) system labeling during semantic and relatedness analysis and (2) expert user labeling while creating the

dataset. Each element of the matrix can be one of the following: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Table 5 displays the confusion matrix of the system. It was generated based on user opinions collected during the data collection stage and inferences made by the reasoner during semantic and relatedness analysis.

**Table 5.** Confusion matrix of the semantic and relatedness system

| | | **Inferred label** | | |
|---|---|---|---|---|
| Actual label provided by the user | | YES | NO | **Total** |
| | YES | 1145 (TP) | 232 (FN) | **1377** |
| | NO | 50 (FP) | 519 (TN) | **569** |
| | Total | 1195 | 751 | **1946** |

- TP represents the number of hyperlinks which are considered useful by both the user and the proposed system.
- FP represents the number of hyperlinks which are considered noisy by the user and useful by the proposed system.
- TN represents the number of hyperlinks which are considered noisy by both the user and the proposed system.
- FN represents the number of hyperlinks which are considered useful by the user and noisy by the proposed system.

Values of six commonly used performance measures in information retrieval systems are visualized in Figure 12. As illustrated, the proposed approach achieves high levels of accuracy and precision, while maintaining error rate sufficiently low.

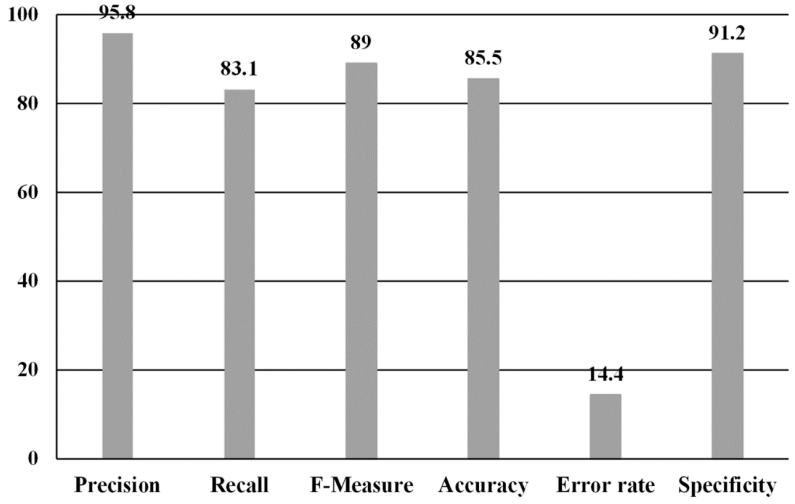


**Fig. 12.** Performance measures of the proposed approach

The extent to which the reasoner uses various semantic and relatedness properties in the DBpedia ontology to represent semantic and relatedness similarities between the Subject and the Object is shown in Figure 13.

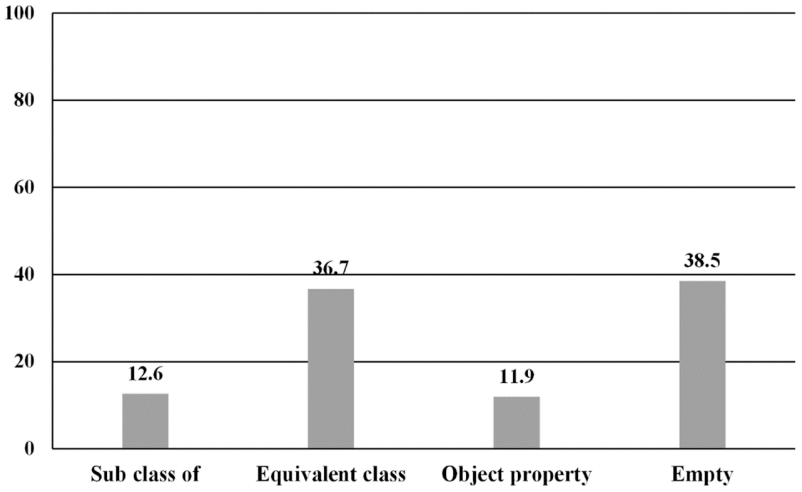


**Fig. 13.** Percentages of properties from the DBpedia ontology used by the reasoner

Figure 13 shows that the reasoner uses the 'Subclass Of' property in 12.6% of cases to relate the source hyperlink concept to the target page concept. The corresponding value for 'Equivalent Class' is 36.7%. These properties indicate semantic similarity between the source and target concepts. Additionally, in 11.9% of cases, the reasoner uses 'Object Property', which represents relatedness similarity. For the remaining 38.5 percent, the DBpedia ontology does not identify any semantic or relatedness properties. Table 6 provides an overview of the results obtained from the semantic and relatedness analysis.

**Table 6.** Semantic and relatedness analysis results

| Number of crawled links | Number of links in the dataset |
|---|---|
| **2665** | 1946 |
| Number of domains | **Ontology classes** |
| **114** | 312 (38%) |
| Useful domains | |
| **Wikipedia, bbc, Facebook, YouTube, eBay** | |
| Noisy domains | |
| **My daily fun dose, Google, full movies free download, Hollywood life, Wikipedia** | |
| Useful domains | **Noisy domains** |
| **61.4%** | 38.55% |
| Target concepts of a noisy link | |
| **Shopping Mall, Currency, Film, Actor, Drug** | |

The feature extraction step retrieves and stores a limited amount of information related to the hyperlink domain, specifically the href values of the <a> tag. Subsequently, the user assesses whether the hyperlink is noisy upon encountering it on the source page. Our goal is to compare our findings with the opinions of the users.

In order to gain a better understanding of the results of the semantic and relational analysis presented in Table 6, we also conduct a comparison with the user's perspective. Table 7 presents an overview of the user's perspective on the usefulness or noise of hyperlinks.

**Table 7.** Analysis of the data collection from the perspective of the user

| Number of crawled links | Number of links in the dataset |
|---|---|
| **2665** | 1946 |
| Number of domains | |
| **114** | |
| Useful domains | |
| **Wikipedia, bbc, Facebook, YouTube, eBay** | |
| Noisy domains | |
| **My daily fun dose, Google, full movies free download, Hollywood life, songmp3** | |
| Useful domains | **Noisy domains** |
| **70.75** | 29.2% |
| Target concepts of a noisy link | |
| **Shopping Mall, Currency, Film, Actor, Model** | |

Tables 6 and 7 demonstrate the effectiveness of the semantic and relatedness approach. Figure 14 visualizes the percentages of hyperlink types from both perspectives. The proposed approach accurately distinguishes between noisy and useful hyperlinks, similar to an expert user.

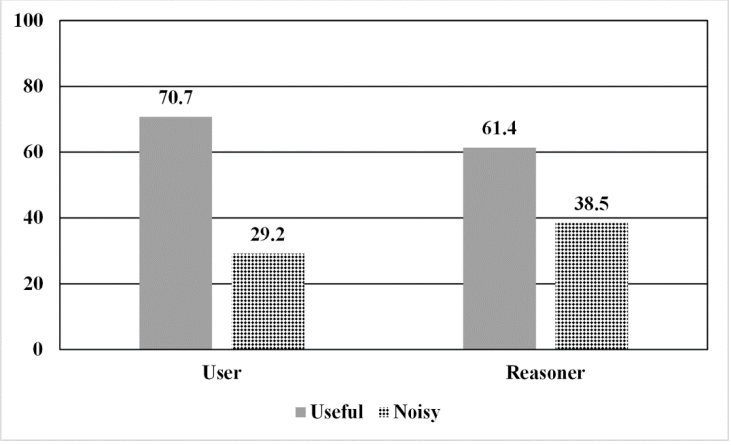


**Fig. 14**. Percentage of hyperlink types as perceived by the user and the reasoned

Figure 14 demonstrates that the reasoner and the user can identify useful hyperlinks in 61.4% and 70.7% of cases, respectively. For noisy hyperlinks, the values are 38.5% and 29.2%, respectively. These results indicate that the reasoner can achieve expert-level results.

### *3.3.1. Scalability*

The final data collection must contain records that can be analyzed through semantic and relatedness approaches, meaning that the records must belong to the classes of the DBpedia ontology. The reasoner needs time to discover new relations, classes, and properties based on the DBpedia ontology and determine the type (i.e. noisy or useful) of the hyperlinks in the data collection. In this section, we discuss the scalability of the reasoner and the entire process. Scalability tests were performed on a computer system with an Intel Core i5 2450M (2.50 GHz) with 4.00 GB of RAM and running a 64-bit operating system. Figure 15 shows the amount of time required to run the semantic and relatedness reasoner for 500, 1000, 1500, and 1946 hyperlinks.

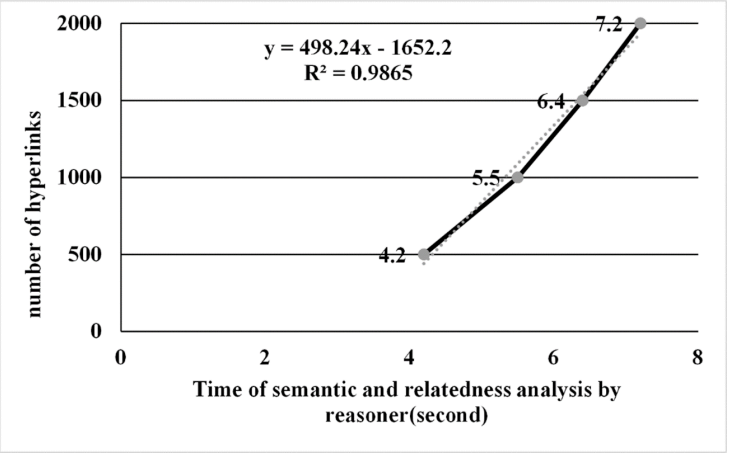


**Fig. 15**. Scalability of the reasoner for different numbers of hyperlinks

The variables 'y' and 'R' represent the number of hyperlinks and the accuracy of the equation, respectively. The reasoner is activated during the initial two to three seconds. Figure 15 assumes that the final dataset is available, while Figure 16 considers the scalability of the entire process from topic identification to semantic and relatedness analysis for the same cases.

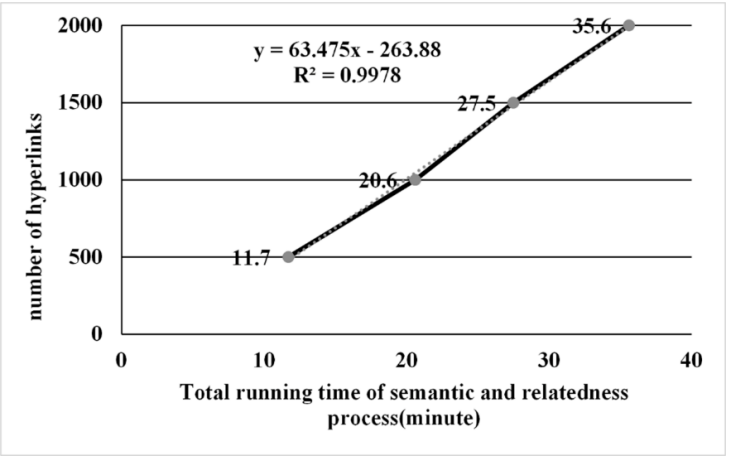


**Fig. 16**. Scalability of the entire system for different numbers of hyperlinks

Our investigations revealed that the matching stage is the most time-consuming step of the process.

## 4. DISCUSSION AND CONCLUSION

Hyperlinks are a type of data that can negatively impact the efficiency of many information retrieval algorithms due to their noise. Most algorithms focus on the string or graph structure of hyperlinks, which can lead to the incorrect removal of useful hyperlinks and the failure to detect noisy hyperlinks in certain cases. This paper examines semantic and relational structures at both the page and site levels. Semantic web technologies, such as ontologies and reasoners, are used to eliminate noisy hyperlinks. A dataset of hyperlinks is created in a separate process and analyzed for both semantics and relatedness. The proposed system distinguishes between noisy and useful hyperlinks using the constructed dataset

as input. The dataset comprises rows that include the class mapped from the hyperlink context topic of the source page, the class mapped from the topic of the target page, a field indicating the noisy or useful nature of the hyperlink from the user's perspective, and the domain name of the source page. The results are then compared to those of the

user to demonstrate the extent to which each semantic or relational property in the ontology contributes to a hyperlink being identified as either noisy or useful. We were able to identify the categories of queries that lead users to noisy hyperlinks, as well as the domains with the highest number of noisy hyperlinks. In addition, our experiments have shown that semantic web technologies are accurate, effective, and scalable in eliminating noisy hyperlinks. Future directions for this work include:

1. Combining the DBpedia ontology with another ontology to cover a larger domain at the T-Box level concepts.
2. Using available datasets on linked data to cover A-Box level concepts.
3. Extending the noise detection operation to include hyperlinks pointing to images, videos, and audio files.
4. Applying various algorithms to match topics to ontology classes.
5. Using semantic tools to identify the topic of hyperlinks and target pages via semantic properties in pages that are constructed using Web 3.0 techniques.

## CONFLICTS OF INTEREST

The authors declare no conflict of interest.

## REFERENCES

[1] Johnson, M. R., & Rodriguez, C. A. (2022). Unveiling Noisy Hyperlinks: A Semantic Analysis Framework. International Journal of Information Retrieval, 30(4), 487–502.

[2] Park, S., & Kim, H. (2021). Exploring the Impact of Noisy Hyperlinks on User Satisfaction: A Case Study. Journal of Human-Computer Interaction, 18(3), 215–230.

[3] Chen, X., & Wang, Q. (2020). Semantic Enhancement of Hyperlink Relevance: An Approach for Noise Reduction. IEEE Transactions on Computational Intelligence and AI in Games, 12(5), 789–802.

[4] Keller, M., & Nussbaumer, M. (2011, September). Beyond the web graph: Mining the information architecture of the WWW with navigation structure graphs. 2011 International Conference on Emerging Intelligent Data and Web Technologies. Tirana, Albania. doi:10.1109/eidwt.2011.2

[5] Kunder, M. d. (2015). The size of the World Wide Web (The Internet) Retrieved from http://www.worldwidewebsize.com/

[6] Wu, Y., Wu, Y., Liu, Y., & Shi, T. (2022, March). The research of the optimized solutions to Raft consensus algorithm based on a weighted PageRank algorithm. 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML). Presented at the 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML), Hangzhou, China. doi:10.1109/cacml55074.2022.00135

[7] Ercan, G., & Cicekli, I. (2007). Using lexical chains for keyword extraction. Information Processing & Management, 43(6), 1705–1714. doi:10.1016/j.ipm.2007.01.015

[8] Samanta, D., Dutta, S., Galety, M. G., & Pramanik, S. (2022). A Novel Approach for Web Mining Taxonomy for High-Performance Computing. In Cyber Intelligence and Information Retrieval (pp. 425–432). Singapore: Springer.

[9] Qi, X., Nie, L., & Davison, B. D. (2007). Measuring similarity to detect qualified links. Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web. Presented at the AIRWeb'07: AIRWeb'07, Third International Workshop on Adversarial Information Retrieval on the Web, Banff Alberta Canada. doi:10.1145/1244408.1244418

[10] Wookey, L., & Geller, J. (2004). Semantic hierarchical abstraction of web site structures for web searchers. Journal of Research and Practice in Information Technology, 36(1), 23–34.

[11] Carvalho, -Da Costa, Chirita, A. L., De Moura, P.-A., Calado, E. S., & Nejdl, P. (2006). Site level noise removal for search engines. In Paper presented at the Proceedings of the 15th international conference on World Wide Web.

[12] Chakrabarti, S. (2001). Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In Proceedings of the 10th international conference on World Wide Web.

[13] Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet:: Similarity: measuring the relatedness of concepts.

[14] Oguz, R. F., Oz, M., Olmezogullari, E., & Aktas, M. S. (2022). Extracting information from large scale graph data: Case study on automated ui testing. In European Conference on Parallel Processing (pp. 364–375). Cham: Springer.

[15] Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., & Stein, L. A. (2004, 12 November 2009). OWL Web Ontology Language. Retrieved from http://www.w3.org/TR/owl-ref/

[16] Wu, B., & Davison, B. D. (2005). Identifying link farm spam pages. Special Interest Tracks and Posters of the 14th International Conference on World Wide Web - WWW '05. Presented at the Special interest tracks and posters of the 14th international conference, Chiba, Japan. doi:10.1145/1062745.1062762

[17] Elakkiya, E., & Selvakumar, S. (2022). Stratified hyperparameters optimization of feed-forward neural network for social network spam detection (SON2S). 1–20.

[18] Solanki, S., Verma, S., & Chahar, K. (2022). A Comprehensive Study of Page-Rank Algorithm. In Evolution in Computational Intelligence (pp. 1–10). Singapore: Springer.

[19] Manning, C. D., Raghavan, P., & Schutze, H. (2012). Introduction to Information Retrieval. doi:10.1017/cbo9780511809071

[20] Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '95. Presented at the the 18th annual international ACM SIGIR conference, Seattle, Washington, United States. doi:10.1145/215206.215333

[21] Lott, B. (2012). Survey of Keyword Extraction Techniques. UNM Education.

[22] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. The Journal of Documentation; Devoted to the Recording, Organization and Dissemination of Specialized Knowledge, 60(5), 503–520. doi:10.1108/00220410410560582

[23] Mccandless, M., Hatcher, E., & Gospodnetic, O. (2010). Lucene in Action: Covers Apache Lucene 3.0. Manning Publications Co.