An Intelligent System for Focused Crawling from Big Data Sources

Ida Bifulco[a], Stefano Cirillo[b,*], Christian Esposito[b,*], Roberta Guadagni[c], Giuseppe Polese[b]

[a] *University of Napoli "Federico II", Department of Physics "E.Pancini", Via Cinthia 21 80126 Monte Sant'Angelo Napoli*
[b] *University of Salerno, Via Giovanni Paolo II 132, 84084 Fisciano (SA)*
[c] *Open Reply, Via Giorgione 59, 00147 Roma*

Abstract

Nowadays, the proper management of data is a key business enabler and booster for companies, so as to increase their competitiveness. Typically, companies hold massive amounts of data within their servers, which might include previously offered services, proposals, bids, and so on. They rely on their expert managers to manually analyse them in order to make strategic decisions. However, given the huge amount of information to be analysed and the necessity of making timely decisions, they often exploit a small amount of the available data, which often does not yield effective choices. For instance, this happens in the context of the e-procurement domain, where bids for new calls for tender are often formulated by looking at some past proposals from a company. Driven by an extensive experience on the e-procurement domain, in this paper we propose an intelligent system to support organisations in the focused crawling of artefacts (calls for tender, BIMs, equipment, policies, market trends, and so on) of interest from the web, semantically matching them against internal Big Data and knowledge sources, so as to let companies analysts make better strategic decisions. The novel contribution consists of a proper extension of the K-means algorithm used by a web crawler within the proposed system, and a semantic module exploiting search patterns to find relevant data within the crawled artefacts. The proposed solution has been implemented and extensively assessed in the e-procurement domain. It has been successively extended to other domains, such as robot programming, cloud providing, and several other domains. Since to the best of our knowledge in the literature do not exists similar systems, in order to prove its effectiveness we have compared its crawling component against similar crawlers, by plugging them within our system.

Keywords: Big Data analytics, Focused crawling, Intelligent system, Natural language processing, Data Clustering, Big Data visualisation

1. Introduction

Within the current data-driven era and the fourth industrial revolution, most companies and public administrations produce huge volumes of data as a result of their ordinary activities. Data are generated by many different devices around us: mobile devices, remote sensing, software logs, cameras, and so on, generating an exponentially increasing volume of data (Hilbert & López). Thus, companies and public administrations aim to develop the capability to analyse such data in order to infer novel knowledge and use it to improve their efficiency, productivity, and competitiveness.

According to McKinsey's report in (Manyika, 2011), the proper management and exploitation of Big Data can pave the way for an important growth of the world economy and the citizens' satisfaction w.r.t. their public administrations. For instance, the European community has estimated that the use of knowledge discovery from Big Data could potentially reduce the expenditure of the European administrative activities,

*Corresponding authors
Email addresses: ida.bifulco@gmail.com (Ida Bifulco), scirillo@unisa.it (Stefano Cirillo), esposito@unisa.it (Christian Esposito), guadagniroberta@gmail.com (Roberta Guadagni), gpolese@unisa.it (Giuseppe Polese)

Published in: Expert Systems with Applications journal.
This version of the article has been accepted for publication, after peer review and is subject to Elsevier terms of use.
The Version of Record is available online at: https://doi.org/10.1016/j.eswa.2021.115560

increasing the generated value from 223 to 446 billion, or even more. To this end, the international Open Government Partnership[1] and the Open Data Charter[2] have been issued, and many countries have joined them. Among these, Italy has been one of the pioneer countries and has consequently issued a strategy for open data in public administrations to meet the demand of civil society to improve the quality and availability of information, to strengthen transparency, and to encourage the reuse of released data[3]. More specifically, the Digital Administration Code[4] demands that data related to the public administrations be freely available according to terms of a license or a regulatory provision, and be accessible throughout information and communication technologies (Carloni, 2005). In addition, to reduce the expenditure of administrative activities within public sectors, full access is provided to the functions of the public administrations and their related data/documents through the Web. A concrete example of such a trend is represented by the e-procurement area, which relies on calls for tender, whose artefacts are easily accessible within the Web by means of the web sites of the public administration or the web site of the Italian government gazette, so that proposals to a given call can be submitted electronically by means of certified emails.

As in other domains, the volume of public procurement data is an extremely large one, since it is typically carried out at all levels of the public administration. By considering the case of the Italian procurement, we can witness a pool of over 30,000 contracting authorities, including national ministries, national agencies, and publicly-owned companies. Public procurement has been progressively digitalised, so that more and more calls for tender are daily published within the web, and concern several different scopes, spanning from building facilities to their maintenance or revamping. However, searching within big data is an extremely complex task, especially for operators without computer science skills, hence unable to properly exploit complex and sophisticated search languages. Another challenging aspect concerns the visualisation of the inferred knowledge, so as to enhance the capability of a human operator to grasp variable insights. Therefore, the ability to carry out timely analysis of the data and display results are crucial points to which researchers are devoting considerable efforts. Almost every big company and public administration are increasingly investing money in data-mining and data-visualisation projects, whose findings are key enablers for those applications that can be used to improve the quality of life in today's society. The application of these concepts is particularly challenging within the context of the call for tender management in public procurement. Moreover, even when calls are published through digital documents, not all of them are fully structured, and even the structured ones do not always abide by the same format or schema, since many different characterisations can be used by public administrations. This represents a considerable burden for private organisations willing to re-use their queries across multiple sources sharing calls for tender. As an example, each Italian public administration provides a customised search system and a different data format, despite each call shares a similar structure imposed by the National Anti-Corruption Authority (ANAC).

In general, after identifying a set of calls for tender of interest, a company needs to quickly come up with its own bid meeting the requirements expressed in a call. To this end, it would be highly desirable to exploit past experiences and lessons learned from participation in past similar procurement opportunities. Although the experience of employees concerning past projects is of pivotal importance, relying on human experts is not always convenient, since they might make decisions based on their intuitions, but without systematically analysing the characteristics of past projects and the companies current workload. On the contrary, the participation in past calls for tender provides huge volumes of artefacts that can be analysed by means of Big Data technologies, in order to select those calls more similar to the current one, which could provide precious hints to prepare a successful bid.

Starting from an extensive experience in the e-procurement domain, our research has focused on two problems that are common to other application domains beyond e-procurement, such as software development, cloud providing, office supplies, and so on: i) crawling artefacts from the web whose informative content matches specific topics of interest, trying to overcome possible linguistic ambiguities of contents written in natural language; ii) matching the characteristics of crawled artefacts against data and knowl-

[1] https://www.opengovpartnership.org/
[2] https://opendatacharter.net/
[3] http://open.gov.it/wp-content/uploads/2017/06/addendum-en.pdf
[4] https://www.agid.gov.it/en/argomenti/digital-administration-code

edge stored within local sources. In order to solve them, we have investigated several big data, machine learning, and natural language processing techniques, some of which have been extended and adapted. For instance, we have derived an extension of the K-means clustering algorithm aiming to prevent its convergence to local minima. In fact, in the experimental section we have shown that our proposed extension does not fall into a local optimum, also shown in (Bifulco & Cirillo, 2018). Such techniques have been implemented in the intelligent system Crawling Artefacts of Interest and Matching them Against eNterprise Sources (CAIMANS).

CAIMANS is composed of several modules. The first one is a novel web crawler capable to extract and pre-process unstructured data from the web. The output of such component is formatted in a suitable way to enable further analysis against a set of query terms, so as to find the artefacts that are more pertinent to the enterprise goals and capabilities. The second module relies on enterprise data and knowledge sources. A third system's module aims to find semantic matches between the crawled artefacts and the knowledge stored within the enterprise sources. Finally, the last module is responsible for visualising the crawled artefacts. These modules can also work in isolation, thanks to a set of RESTful web services and a proper Graphical User Interface. The approach underlying CAIMANS has also been conceived to work off-line, by running the analysis in batch mode and visualising the results at the most convenient time for the user. To this end, we have also conceived a dynamic page detection approach, in order to remove the contents of dynamic pages from the answer set, since their contents might no longer be pertinent at a time when crawled artefacts are analysed.

In the context of the e-procurement domain, CAIMANS has been experimentally used to support two phases of the call for tender management process: crawling new calls from the web and searching the enterprise data and knowledge sources to extract useful information from past calls and corresponding bids, in order to support the preparation of a suitable proposal for the call for tender being examined. The aim has been to support a human operator in the overall process of formulating a competitive response to some calls of interest in a relatively short time and in more an effective way, by leveraging on positive and negative past experiences stored within the company's data and knowledge sources.

The key contributions of our proposal are the following ones:

- An extension of the K-means clustering algorithm, relying on multiple random starting points (Bifulco & Cirillo, 2018), in order to tackle the well-known drawback of K-means to converge to a local minimum, since experiments revealed this to be particularly critical in the surveyed domains. Based on this enhanced version of K-means, we have built an advanced crawler capable of effectively segmenting web search results into clusters, increasing the capability of end-users to analyse them.

- A semantic module relying on natural language processing techniques and the cosine similarity to analyse the contents of the crawled web pages and extract information that is pertinent to a domain of interest for the company. This module also includes a customizable component exploiting search patterns to select the most relevant data, aiming to show them through a structured representation.

- An extended validation, accomplished in cooperation with industrial stakeholders from the e-procurement domain, on real size use cases, and comparative analysis with respect to existing systems, on several application domains, proving the effectiveness and usability of CAIMANS.

The paper is organised as follows. Section 2 contains the description of the relevant state of the art on the addressed topic and challenges. Section 3 presents the case study of call for tender search and the way CAIMANS has been exploited to solve typical problems of this domain. In section 4, we describe the assessment we have conducted on CAIMANS. Section 5 concludes the paper by presenting our conclusions and future directions.

2. Problem Statement and Analysis of the Related Works

The main goal of CAIMANS is to crawl contents from the web, supporting the selection of contents of interests, and matching them against the contents of the enterprise data and knowledge sources. Thus,

in this section existing semantic search engines and crawlers are mainly surveyed and analysed. Generally speaking, a search engine operates through three main processes: scanning (crawling) the Web, indexing, and ranking (sorting) the results obtained from a crawler, and visualising results. A web crawler is a bot (program or script that automates operations) that periodically traverses the Web, following the links and the strings contained within the visited Web pages. In particular, starting from a set of Web pages, called the seed set, a crawler follows the links contained in them. Given the current size of the Web, even powerful search engines cover only a portion of the contents publicly-available over the Internet. Since a crawler typically downloads only a fraction of the Web pages, it is highly desirable that such a fraction contains the most relevant pages and not only a random sample of the Web. Generally, the crawler should embed proper heuristics to focus the exploration of those portions that are highly likely to contain the data of interest.

Research on crawling heuristics to evaluate the contents of retrieved Web pages is gaining increasing attention from researchers, and previous experiences are available in (Yuvarani et al., 2006; Kumar & Vig, 2013; Pinnamaneni, 2013; Dhingra & Bhatia, 2015). A recent study has proposed a new methodology for web crawling, based on reinforcement learning, which considers the distance between two pages as the average number of clicks to reach one page from the other (Bidoki & Yazdani, 2008), also performing a comparison w.r.t. most relevant ranking algorithms. Other recent studies have analysed main issues and limitations of the ranking algorithms PageRank and SALSA (Lempel & Moran, 2000), also providing new variants to them, aiming to optimise their ranking methodologies (Goel et al., 2019; Lozano & Calzada-Infante, 2019). These algorithms are generally used to evaluate a large number of pages whose topics might not be related to those of the search query. To this end, new methodologies must be devised to minimise off-topic results and maximise search-query related results, starting from the crawling phase.

A recent crawler exploits the geographical position of the user browser, web portal, web servers, and original web pages, to improve the quality of the extracted pages (Cambazoglu et al., 2007). Although this strategy can improve the overall quality of results, this type of crawler requires a deep and costly analysis of every single page, due to the fact that it considers several hidden parameters that may not be available.

In (Jain et al., 2013) the authors propose an approach to design the architecture of a web-crawler able to analyse the extracted data using a classic version of the K-means algorithm. However, as shown in (Vassilvitskii & Arthur, 2006; Vattani, 2011), there are several cases in which the classical K-means algorithm cannot guarantee the achievement of the global optimum. For this reason, the proposed search engine exploits an optimised version of the K-means algorithm able to analyse the search space moving away from the local minima (Bifulco & Cirillo, 2018).

Similarly to the approach defined in (Balbi et al., 2018), the proposed crawler prioritises the URLs in the queue, which allows the search engine to visit "important" pages first. Nevertheless, by adopting the *Focused Crawling* strategy, the proposed crawler limits the Web exploration to the pages that are relevant to some pre-defined topics (Chakrabarti et al., 1999). Target descriptions in focused crawling are dependent on the various applications where the web crawler has been adopted. Indeed, a large variety of focused crawlers has been created, aiming to extract specific information from the web. Several studies show the application of the focused approach in different contexts: extracting hidden information from the Deep Web (Hernández et al., 2019; Al-Nabki et al., 2019), searching for educational materials (Premlatha & Geetha, 2011), or for discovering, annotating, and classifying biomedical information (Dong & Hussain, 2010).

Generally speaking, in user-centric crawling (Pandey & Olston, 2005) the targets are mined from user queries to guide the refreshing schedule of a generic search engine, whereas in (Vidal et al., 2006) the target is described by the DOM tree of a manually selected sample page, and the crawling is limited to specific Web sites.

Recent works focus on content-based duplicate detection, whereby Web documents are first characterised through some fingerprints, such as Shingles (Broder et al., 1997) or SimHash (Manku et al., 2007), which are pairwise compared by means of L_2 or Hamming distance to detect duplicates. However, content-based de-dup can only be carried out offline, after Web pages have been downloaded. A recent work accomplishes URL-based duplicate detection, by mining rules for URLs sharing some similar text (DUST) (Schonfeld et al., 2006). However, such rules are extremely risky in practice, since URL formulations are too multifarious to generate robust rules. Furthermore, such a method still needs to analyse logs from the target Web server or some previous crawling process.

An industry-related work is the Sitemaps Protocol[5], which represents an XML file listing URLs and additional information, such as update frequencies. However, it is hard to maintain such a protocol file for the Web, as its content continuously changes.

A classic approach to semantic search leverages on domain ontologies (McDaniel & Storey, 2019), in order to model the semantics of terms and entities belonging to a given domain, by indicating their attributes and relationships, so as to expand the meaning of a particular concept provided as input for a given query. Domain ontologies are commonly used in enterprise tools to achieve semantic capabilities, but they also suffer from several drawbacks. Basically, when a domain ontology is missing, it is extremely difficult to turn specialised domain knowledge (contained in textual artefact or domain experts) into ontologies, as it demands abstract and effective concept representations. In fact, domain knowledge is typically difficult to understand and contains many contradictions or misinterpretations. Moreover, ontologies fail to perfectly represent synonymies, and the existing ontology building tools exhibit quite a heavy learning process, making the construction of an ontology a time-consuming and troublesome activity for users without proper computer science backgrounds. Since these are the type of users targeted in our project, we have based our system on solutions not relying on ontologies.

3. The proposed intelligent system

Current Web search engines require users to search for artefacts of interest by mainly entering query strings. Generally, this limits the search and does not guarantee that correct results will be obtained immediately. Human experts must carry out many manual searches in order to obtain useful results. In fact, often within the Search Engine Results Page (SERP) there are many pages outside the search scope. To this end, CAIMANS can reduce the overall search time, by increasing the number of correct results w.r.t. manual search methodologies. In particular, thanks to the advanced management of search parameters and URLs to avoid, CAIMANS can guarantee a faster convergence of the search engine towards a set of artefacts of interest. For instance, in the calls for tender domain, CAIMANS enables a company to extract calls from the Web and classify them according to the company's needs and past experiences. At the same time, CAIMANS can search company's data and knowledge sources to retrieve artefacts related to past calls for tender, focusing on the proposed solutions and achieved evaluations, so as to have a starting point for writing a proposal for the current calls for tender. This will potentially reduce the effort for preparing proposals, yielding cost reduction, and increasing the chances of a company to gain contracts than relying only on human experts.

In this work, we will show that CAIMANS improves manual search in terms of search time and quality of extracted results in several application domains. The following sections will explain the solutions underlying CAIMANS, together with the general design objectives, requirements, and its modular architecture.

3.1. System Architecture

CAIMANS has been conceived as a modular platform, in which each component can be singularly used and interact with other ones by defining a proper workflow. Each component was developed independently from the others, by formally specifying its RESTful interface. Figure 1 shows system architecture of CAIMANS, in which we have detailed the interactions between its modules. The following sections show the characteristics and techniques used for the development of the individual components.

3.1.1. Crawler Module

Such a module has the duty of traversing the Web and returning the artefacts pertinent to a given query, by following exploration and priority rules and abiding by search limits. In particular, this module represents a focused crawler, which is based on Linkage Locality criteria (Kumar et al., 2017). In other words, starting from a set of user-defined web pages that are pertinent to the searched topic, the exploration begins by retrieving all the pages related to them. This is based on the assumption that the web pages on a

[5]https://www.sitemaps.org/protocol.html

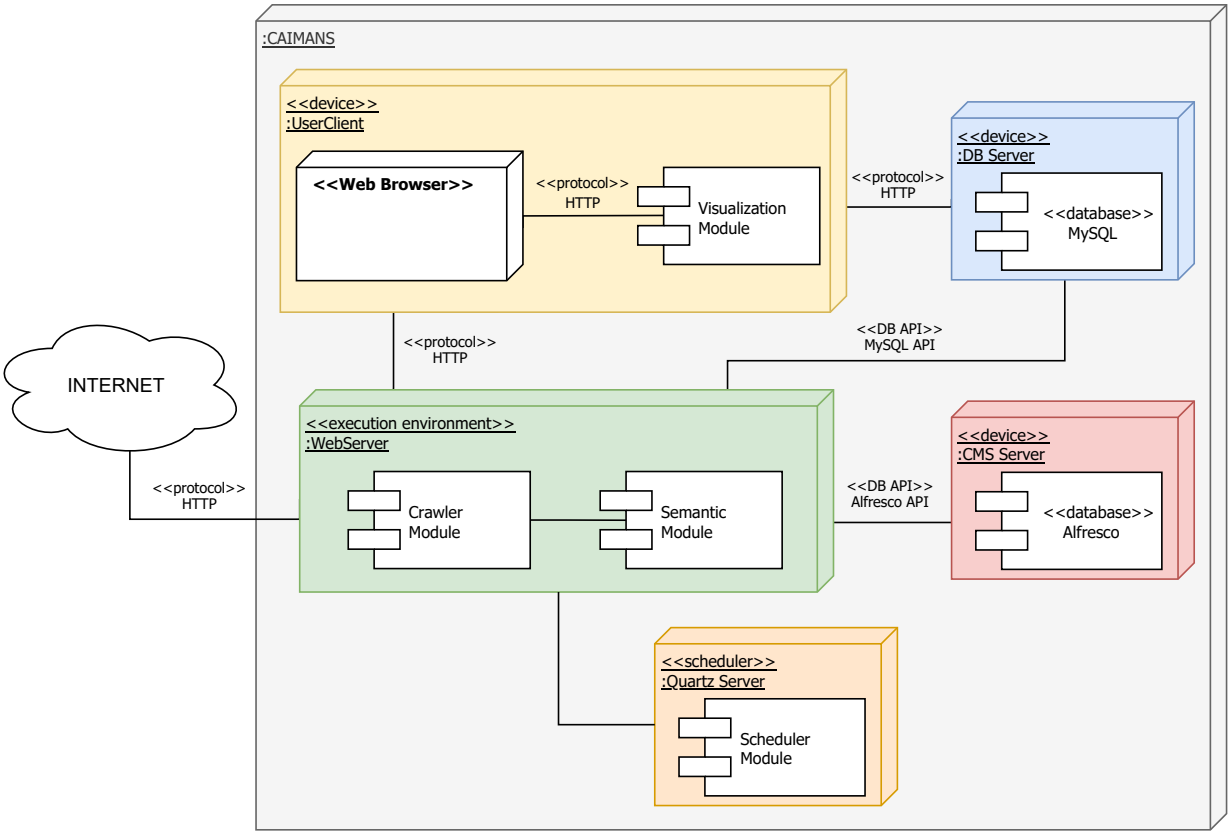


Figure 1: The architecture of CAIMANS.

given topic are more likely to link to those concerning the same topic. For example, if the crawler analyses the content of the Italian government gazette web site, most linked pages contain information about calls for tender. These links are placed in the exploration queue one after the other, in order to be analyses by different semantic levels. To specify the priority order in which URLs have to be visited during the crawling phase, a navigation queue has been defined. Furthermore, to define search limits, the crawler uses a customised URLs blacklist that excludes out-of-context domains from scanning.

The crawler module requires the following further functional components:

- A component to download web pages from the absolute URLs using the HTTP protocol;
- A component for extracting content and links from HTML documents;
- A component to validate the syntax and the existence of a URL;
- A component for determining whether a URL has been encountered before;
- A component for extracting content and links from RSS Feed;
- A component to avoid the exploration of blacklisted domains/ pages.

Moreover, the whole crawler module has been realized by extending the web crawler Mercator architecture (Heydon & Najork, 1999). The latter enables a focused search, ensuring that all components are independent from each other and cooperate in a single system by taking input data from the previous component, processing it, and returning the output to the next component. All the components can be maintained and updated separately from each other, reducing the impact of changes in the other modules. Crawling is

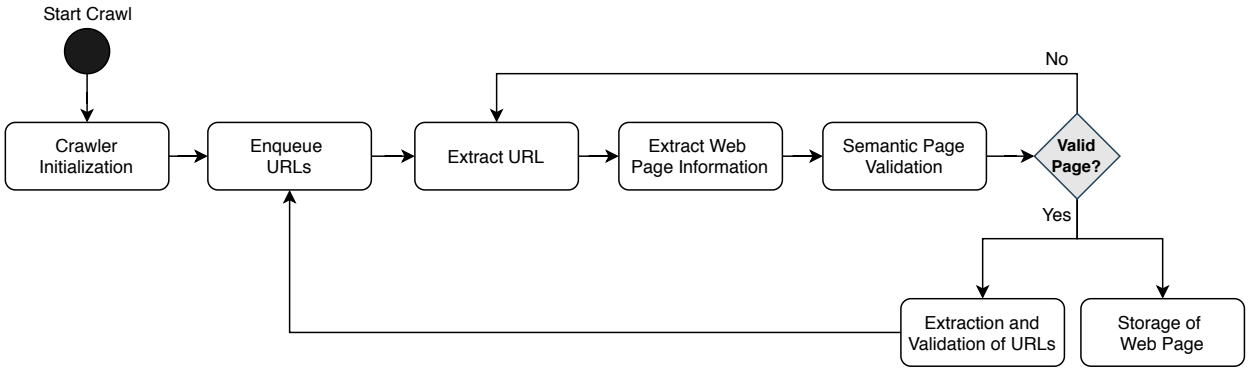


Figure 2: Flowchart of Crawler Module.

accomplished through a scheduled sequential process managed by the Quartz Scheduler (Cavaness, 2006), through a CRON expression[6]. The latter is a string composed of 7 space-separated fields, representing seconds, minutes, hours, day, month, weekday, and year, respectively.

Example 1. If we consider the CRON string "0 15 10 ? * MON-FRI", the schedule fires at 10:15 am every Monday, Tuesday, Wednesday, Thursday, and Friday. The value "*" is used to select all values within a field, whereas the value "?" is useful when it is not necessary to specify the day.

The scheduled search job is automatically started at a given time, or based on a recurring schedule. These kinds of tasks can be easily performed on external servers or cloud machines. The scalable architecture of Mercator makes is it possible to parallelize the single components according to the search engine execution platform. The execution steps of the crawling module are shown in Figure 1.

Execution steps. The first step corresponds to the crawler initialisation, consisting in the definition of the query string, the keywords, and the maximum number of pages that the crawler must explore in order to reduce the number of incorrect results and to increase search accuracy. The next crawling phase starts when an absolute URL is extracted from the URL queue. In order to avoid that the crawler only analyses pages from the same domain, the elements in the queue are shuffled after n loops. The parameter n is an integer randomly generated in the range K/2 and K, where K is the maximum limit of pages to be explored. Next, the frontier component of Mercator extracts the URL, checking whether it is not contained in a blacklist. Each web page is analysed by using an HTML analyser, based on the XPath syntax. The latter enables fast access to the content of the HTML tags. Thus, it avoids processing a complex HTML parser. A stemming algorithm is used to process all words in the text, reducing those with the same root to a common format, by stripping the derivational and inflectional suffixes from each word. More specifically, we have employed a stemming algorithm for the Italian language (De Souza, 2012), that allows the module to quickly analyse the text extracted from the given web page by means of Natural Language Processing (NLP) techniques. To this end, we have used basic NLP techniques, since the volume of artefacts to be processed is quite large and using complex techniques would not allow to have a fast processing. Furthermore, we have extended this algorithm with a module capable of using a set of keywords to be involved in the analysis of web pages. Once the keywords have been processed, each word and its synonyms are searched in the text of the web page to verify the correctness of results.

Example 2. If we consider the keyword "fixtures", by using the stemming algorithm its root "fixtur-" is extracted. Starting from the singular form of the word, all of its related synonyms are searched, i.e., "doors", "windows", "shutters", and so on. For each found synonym, its root is extracted, i.e., "door-", "window-"", "shutter-", and searched within the text.

[6]http://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/tutorial-lesson-06.html

The extraction module uses accurate regular expressions to extract the contents of the calls for tender, i.e., prices, opening date, closing date, SOA categories, and other information useful for classifying the results. Furthermore, in addition to the regular expressions already defined, it is possible to define new expressions to be involved in the search. This type of strategy ensures that each module is fully customizable during the initialisation phase. Towards the end of the cycle of phases, a test is performed to verify whether the page is valid. If so, then the crawler stores the results in a database, it extracts the links contained in it, storing them in the exploration queue, and it goes back to its second phase. During this process, dynamic URLs related to servers or main pages of generic portals are discarded, in order to avoid that pages out of context are inserted in the exploration queue. Moreover, in the validation phase, the crawler checks whether a link is an absolute URL, and if it not refers to a web site already visited before and inserted in the blacklist. The validation component has a single crawl method that takes in input a URL and it returns a boolean value indicating whether or not the URL should be added to the queue. On the other hand, if the page is not valid, the crawler goes back to its third phase. The process stops when the crawler exceeds the time limit or the number of pages to be visited.

3.1.2. CMS Module

This module needs to process several types of artefacts. For instance, civil engineering projects include technical artefacts, describing the design of a building or its maintenance plan, but also administrative ones, stating the financial planning and the project schedule. Such a volume of data is characterised by having multiple kinds of formats, from PDF to DOCS, or a level of structure and formalisation, spanning from unstructured material (such as images from scanned documents) to more structured ones (such as XML rendering of architectural plans). Traditionally, such a set of materials is poorly managed by the companies, which usually maintain flat storage within the folders of their servers or employees' computers. Even though such a strategy does not require a considerable background in computer science (motivating its large usage), it is not efficient due to its intrinsic difficulty in retrieving a specific artefacts of interest within such a set. The traditional approach of using relational databases can help in having more effective retrieval, thanks to queries expressed in a Structured Query Language (SQL). However, such a solution is viable for storing properties, but it is less effective when dealing with files. A Content Management System (CMS) (Paivarinta & Munkvold, 2005) represents a trade-off between folder storage using the operative system and a relational database, so that the artefacts of interest are in the file system, and the CMS manages pointers to them within certain tables coupled with meta-data supporting queries for their retrieval. Such pointers are transparent to the users, and the CMS guarantees their consistency concerning artefact mobility within the file system.

In our project, we have adopted the Alfresco platform[7] for storing and retrieving the artefacts of interest. For instance, in e-procurement domain, the file-system can be structured by considering the different parts of any civil project, each of which can contain artefacts related to a call for tender or a bid for it. This type of strategy allows CAIMANS to quickly interact with company artefacts, aiming to characterise the search by involving information extracted from the company's knowledge base.

We have built a set of RESTful web services on top of Alfresco, in order to enable the storage and retrieval of artefacts. The API used to interact with Alfresco has been the one provided within the Content Management Interoperability Services (CMIS) (Choy et al., 2010) (and the Apache Chemistry [8] library for .NET), so that we can use any possible CMIS-compliant CMS and replace Alfresco based on the customer needs. Such services have been made secure by using the standard JSON Web Token (JWT) (Jones et al., 2015), for stateless authentication and authorisation. The retrieval of artefacts is made possible by using the CMIS Query Language, which is based on the SQL-92 SELECT statement. Such a language has a syntax particularly troublesome for a user with a poor computer science background, and it requires knowledge on the right term to look for, because if the query contains a synonym of a term contained in the artefact the match is not detected. To simplify the query, it is possible to have a different approach by using a

[7] https://www.alfresco.com/
[8] https://chemistry.apache.org/

faceted search, which involves augmenting traditional search techniques with means to enable users narrow down search results thanks to the faceted classification of items. The similarity between an artefact and the provided query can be detected by using Solr[9], a library based on the Apache Lucene, thanks to a term indexing process. Thus, we have used the API provided by the SearchService of Alfresco to let a user retrieve artefacts employing the term-matching provided by CMIS and the faceted search of Solr. The extraction and analysis phases are performed by the semantic module which will be discussed in the next section.

3.1.3. Semantic Module

The purpose of this module is to analyse the results extracted from the crawler module and the artefacts saved in the corporate CMS to define those that are closest to the search parameters and the query string. This can be described as an Information Retrieval problem (Ghorab et al., 2013), in which it is necessary to exploit Query Expansion techniques (Carpineto & Romano, 2012; Caruccio et al., 2017) for minimising the query-results mismatch, therefore improving retrieval performance. In order to realise an effective artefact search, techniques from the NLP literature are usually exploited (Foucault et al., 2011; Sun et al., 2005; Acharya & Parija, 2010; Grosman et al., 2020). They enable to overcome the limits of basic text-matching realised by traditional query languages (such as SQL), returning more pertinent results thanks to stemming and similarity operations. Thus, this module benefits not only from the user's query but also from keywords and their synonyms. Moreover, thanks to the interaction with the CMS module, the semantic module can exploit the historical knowledge of a company to include several new related terms for analysis.

Figure 3 shows the workflow of the semantic module. As we can see, it accomplishes its analysis by considering several phases and techniques. Initially, to determine the artefacts containing the query keywords, the frequency of the keywords contained within each artefact (e.g., call for tender) is calculated through the TF-IDF weighting scheme (Lakshmi & Baskar, 2019). Moreover, other than calculating the keyword frequency, the TF-IDF algorithm (Ramos et al., 2003) also calculates a value that is directly proportional to the frequency of the term in the document, but inversely proportional to the frequency of the term in the entire collection of documents. In this way, common keywords will have a lower value than those appearing less frequently within the artefacts.

Although the use of TF-IDF allows the semantic module to efficiently evaluate the results obtained by the crawler module with respect to the performed search, there are some cases in which this technique is not suitable. Indeed, if there are no common terms between a web page and a given topic, we cannot achieve a proper similarity for the web page. As described in (Du et al., 2015), this problem can lead to ignoring some links to pages that are pertinent, since there must also be common terms among the topics of the hyperlinks in order to achieve a fair similarity value of the artefacts. For this reason, the semantic module combines TF-IDF and cosine similarity in order to evaluate results, by also using the anchor text of hyperlinks as its artefacts. The experimental results performed in (Du et al., 2015), demonstrate that the combination of these techniques improves the performance of focused crawlers, outperforming other focused crawlers relying on different metrics and techniques. More specifically, the cosine similarity has been used to determine the artefacts that are more similar to the query parameters, based on the frequency calculated in the previous phase. This similarity metric has been chosen for its independence from the length of the artefact, whereby artefacts with the same composition but different word counts will be treated identically. Thus, based on the cosine similarity, the retrieved artefacts have been sorted and divided into the following three lists:

1. **White list**: the subset of artefacts of interest to the user;
2. **Black list**: the subset of artefacts that have been selected by the search engine but have no relevance to the context of the search;
3. **Gray list**: the subset of artefacts whose relevance to the user is uncertain.

To derive the threshold values and distribute the artefacts among the three lists mentioned above, an empirical study has been carried out, aiming to minimise the overlap between artefacts.

[9]https://lucene.apache.org/solr/

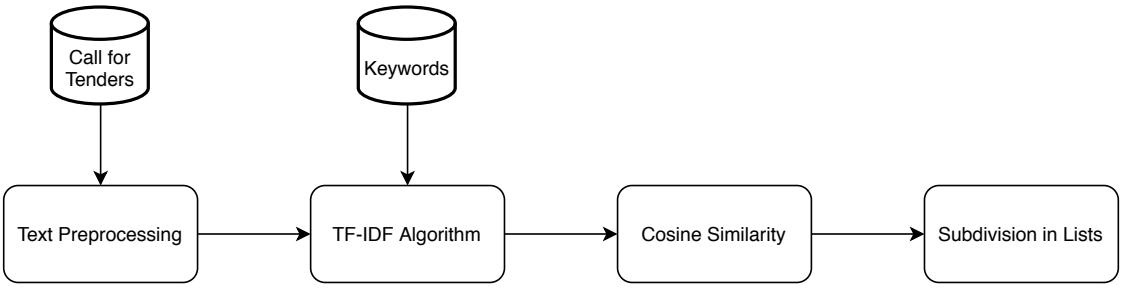


Figure 3: Flowchart of the Semantic Module.

3.1.4. Visualisation Module

This module clusters similar artefacts contained in White and Gray lists, in order to let the user gain immediate insights from retriever artefacts. The goal of cluster analysis is to group the results based on information found in the data describing artefacts and their relationships, so that artefact belonging to the same cluster will be related to one another and unrelated to those in other clusters. The greater the similarity within a group and the greater the difference between groups, the better is the clustering.

This type of analysis plays an important role in several areas, such as social sciences (Langari et al., 2020), documents classification (Arumawadu et al., 2015; Hu et al., 2008; Liu et al., 2013), statistics, pattern recognition, information retrieval, data mining, and so on. However, in some cases cluster analysis is only a useful starting point for other purposes, such as data summarization or data-visualisation. Among the available clustering algorithms, one of the most used and studied is K-means (Kim et al., 2020), which is an unsupervised learning algorithm able to easily adapt itself to several contexts and to quickly analyse large datasets. In particular, it is a popular method of cluster analysis, which partitions a dataset X into K disjoint clusters, such that each element belongs to the cluster with the nearest mean.

Clustering Optimization. One of the drawbacks of the K-means algorithm is that it often convergence to local minima. To tackle this problem, in this paper we propose an empirical solution to extend the search out of the local minimum, aiming to reach minima closer to the global one. In particular, the solution relies on multiple executions of the K-means algorithm with different random starting points. All the obtained solutions are saved and displayed to compare the achieved results. We have tested the proposed extension in the e-procurement domain by using a dataset of calls for tender consisting of 150 calls concerning all the Italian regions, each consisting of 17 features. In order to search the appropriate number of clusters, we used the Elbow method (Gove, 2015). The latter looks at the percentage of variance expressed as a function of the number of clusters. The method relies on the idea that one should choose a number of clusters so that adding another cluster will not yield an improved modelling of the data. The percentage of variance expressed by the clusters is plotted against the number of clusters. Notice that, at some point the marginal drops dramatically, forming an angle in the graph. Thus, the correct number of clusters is chosen at this point. The diagram in Figure 4 show the results of the Elbow method.

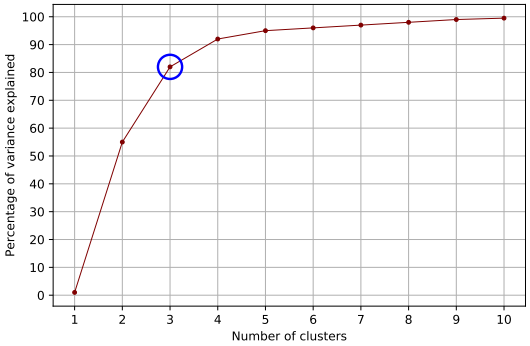


Figure 4: Elbow diagram.

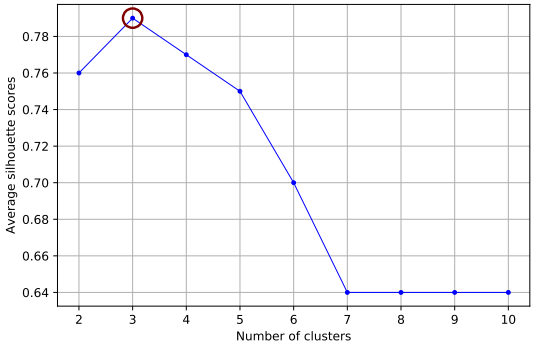


Figure 5: Silhouette diagram.

In particular, this diagram shows that the gain in explained variance changes significantly from 3 to 4 clusters, hence the optimal number of clusters is between 3 and 4. Thus, further tests are necessary to decide weather is the optimal number of clusters for the K-means algorithm is 3 or 4. To this end, we exploit the Silhouette method (Lensen et al., 2017), which defines how similar a point is to its own cluster (cohesion) compared to other clusters (separation). Figure 5 shows the average silhouette scores achieved in our tests. Notice that the silhouette score reaches its global maximum at the optimal K. Thus, we have chosen $K = 3$ as the number of clusters for the K-means algorithm.

Each time the clustering algorithm runs it picks K random seeds to determine the starting centroids of each cluster. In order to guarantee different random seed values, they were generated by using the timestamp with microsecond precision. The algorithm stops the iteration when the distances between consecutive points are less than the given tolerance, which in our case has been experimentally set 0.1. The outputs of each algorithm execution are the coordinates, the labels, and the inertia value of the K cluster centres. In order to visualise the obtained clusters and to reduce the problem dimensions, the Multidimensional Scaling (MDS) algorithm has been used (Borg & Groenen, 2003).

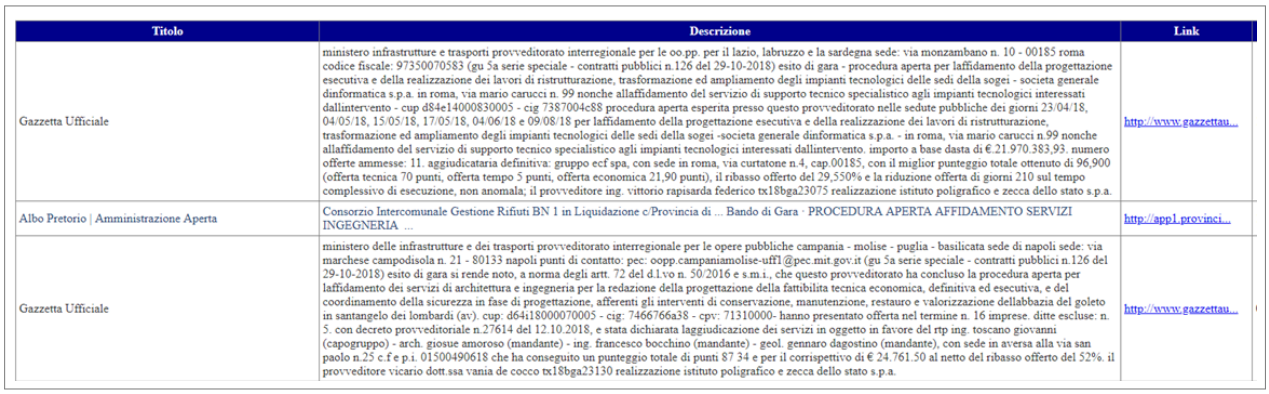
The visualisation module groups the extracted data into a given number K of clusters, according to features describing the artefacts of interest. In particular, in the e-procurement domain we have selected following features of calls for tender: *amounts*, *opening dates*, *closing dates*, and *SOA category*[10]. As said above, according to the Elbow and Silhouette methods we have selected $K = 3$. Moreover, in order to improve the quality of the final classification and to remove possible false-positive results, the visualisation module exploits a dynamic page removal module. For instance, the following are examples of pages considered as dynamic: homepages, pages with a clear IP in the URL, and pages with frequent updates. In this way, all the identified groups are disjointed and their intersection is empty.

4. Experimental Results

The prototype of the proposed system has been developed in *C#* .NET framework 4.7.1[11], based on the Model-View-Controller (MVC) architectural pattern. Each module has been developed standalone and combined into a single .NET solution connected by project references. A RESTful API service has been integrated into the solution to simplify inquires to both the crawler and the CMS.

The screenshot in Figure 6 shows the results produced by CAIMANS for the e-procurement case study. It shows the extracted calls for tender, sorted by the similarity values of results, according to the search parameters. For each result, the table shows the title, a short preview of the artefact, and the referring URL. The background colour is one of the lists to which the document has been assigned. Figure 7 shows

[10] https://www.anticorruzione.it/portal/public/classic/Servizi/ServiziOnline/SocietaOrganismoAttestSOA
[11] https://dotnet.microsoft.com/



Titolo	Descrizione	Link
Gazzetta Ufficiale	ministero infrastrutture e trasporti provveditorato interregionale per le oo.pp. per il lazio, labruzzo e la sardegna sede: via monzambano n. 10 - 00185 roma codice fiscale: 97350070583 (gu 5a serie speciale - contratti pubblici n.126 del 29-10-2018) esito di gara - procedura aperta per laffidamento della progettazione esecutiva e della realizzazione dei lavori di ristrutturazione, trasformazione ed ampliamento degli impianti tecnologici delle sedi della sogei - societa generale dinformatica s.p.a. in roma, via mario carucci n. 99 nonche allaffidamento del servizio di supporto tecnico specialistico agli impianti tecnologici interessati dallintervento - cup d84e14000830005 - cig 7387004c88 procedura aperta esperita presso questo provveditorato nelle sedute pubbliche dei giorni 23/04/18, 04/05/18, 15/05/18, 17/05/18, 04/06/18 e 09/08/18 per laffidamento della progettazione esecutiva e della realizzazione dei lavori di ristrutturazione, trasformazione ed ampliamento degli impianti tecnologici delle sedi della sogei -societa generale dinformatica s.p.a. - in roma, via mario carucci n.99 nonche allaffidamento del servizio di supporto tecnico specialistico agli impianti tecnologici interessati dallintervento. importo a base dasta di €.21.970.383,93. numero offerte ammesse: 11. aggiudicataria definitiva: gruppo ecf spa, con sede in roma, via curtatone n.4, cap.00185, con il miglior punteggio totale ottenuto di 96,900 (offerta tecnica 70 punti, offerta tempo 5 punti, offerta economica 21,90 punti), il ribasso offerto del 29,550% e la riduzione offerta di giorni 210 sul tempo complessivo di esecuzione, non anomala; il provveditore ing. vittorio rapisarda federico tx18bga23075 realizzazione istituto poligrafico e zecca dello stato s.p.a.	http://www.gazzettau...
Albo Pretorio | Amministrazione Aperta	Consorzio Intercomunale Gestione Rifiuti BN 1 in Liquidazione c/Provincia di ... Bando di Gara · PROCEDURA APERTA AFFIDAMENTO SERVIZI INGEGNERIA ...	http://app1.provinci...
Gazzetta Ufficiale	ministero delle infrastrutture e dei trasporti provveditorato interregionale per le opere pubbliche campania - molise - puglia - basilicata sede di napoli sede: via marchese campodisola n. 21 - 80133 napoli punti di contatto: pec: oopp.campaniamolise-uff1@pec.mit.gov.it (gu 5a serie speciale - contratti pubblici n.126 del 29-10-2018) esito di gara si rende noto, a norma degli artt. 72 del d.l.vo n. 50/2016 e s.m.i., che questo provveditorato ha concluso la procedura aperta per laffidamento dei servizi di architettura e ingegneria per la redazione della progettazione della fattibilita tecnica economica, definitiva ed esecutiva, e del coordinamento della sicurezza in fase di progettazione, afferenti gli interventi di conservazione, manutenzione, restauro e valorizzazione dellabbazia del goleto in santangelo dei lombardi (av). cup: d64i18000070005 - cig: 7466766a38 - cpv: 71310000- hanno presentato offerta nel termine n. 16 imprese. ditte escluse: n. 5. con decreto provveditoriale n.27614 del 12.10.2018, e stata dichiarata laggiudicazione dei servizi in oggetto in favore del rtp ing. toscano giovanni (capogruppo) - arch. giosue amoroso (mandante) - ing. francesco bocchino (mandante) - geol. gennaro dagostino (mandante), con sede in aversa alla via san paolo n.25 c.f.e p.i. 01500490618 che ha conseguito un punteggio totale di punti 87.34 e per il corrispettivo di € 24.761.50 al netto del ribasso offerto del 52%. il provveditore vicario dott.ssa vania de cocco tx18bga23130 realizzazione istituto poligrafico e zecca dello stato s.p.a.	http://www.gazzettau...

Figure 6: Results interface.

the three clusters that the visualisation module has identified for each extracted call for tender. Each cluster is represented by a different colour, and the size of the individual circles indicates the similarity value of the retrieved artefact with respect to the query parameters.

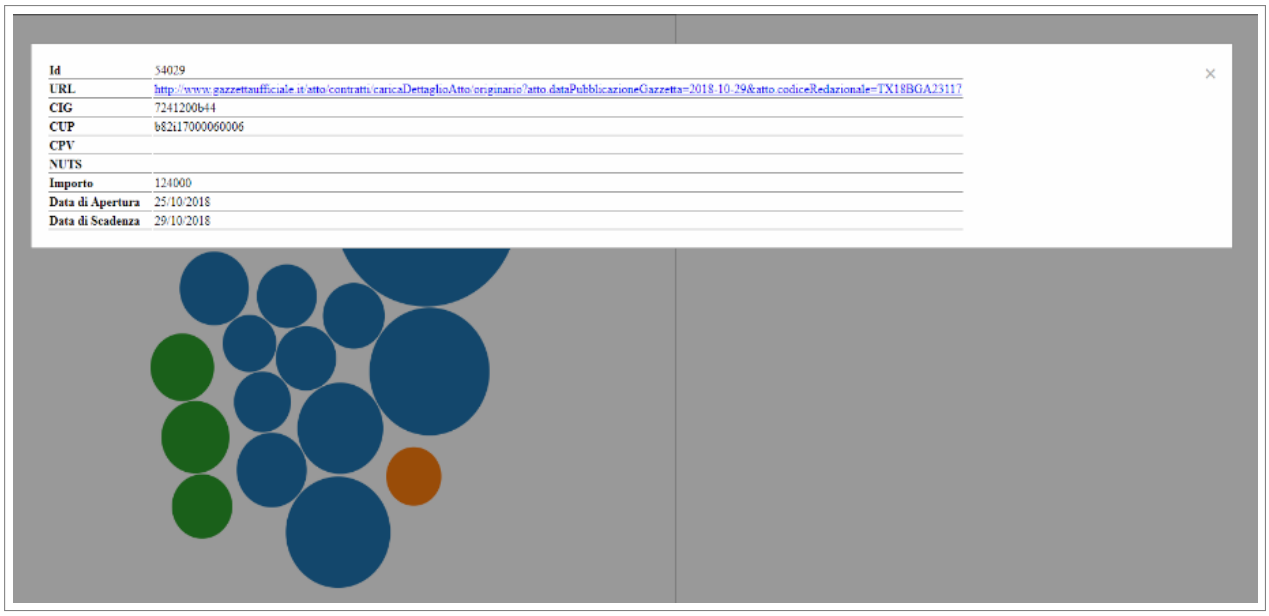


Figure 7: Displaying the result of the clustering phase.

4.1. Evaluation Criteria

The most important measure for evaluating a search engine is the relevance of the retrieved results w.r.t. to the search parameters. Generally, the algorithms underlying web search engines analyse page semantics, and the number of references to the page, aiming to reduce the number of retrieved results and increase their quality.

In this study, before comparing the performances of CAIMANS with those of similar systems, we have first compared it with Google by using precision and recall metrics computed on the set of the retrieved artefacts. To this end, in order to construct the confusion matrix, a domain expert needs to be involved in the evaluation phase in order to define true positives, false positives, and false negative, whereas true negatives were automatically determined by CAIMANS through the blacklist. Since Google does not divide

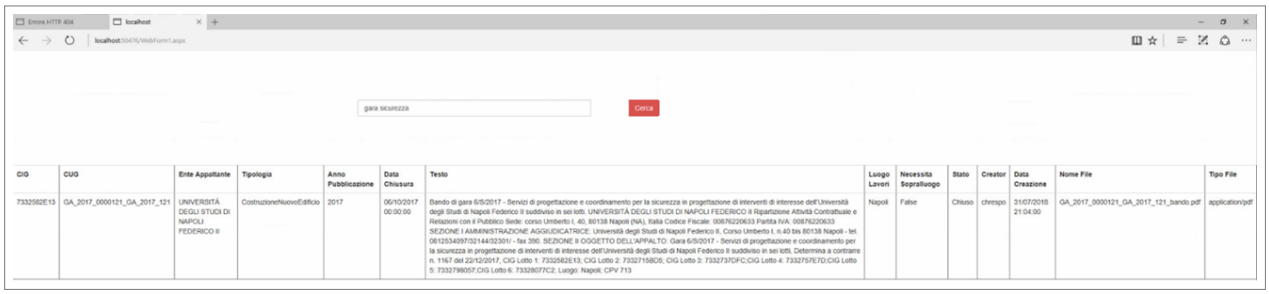


Figure 8: Semantic Search form - results.

results like CAIMANS, rather it retrieves the list of one hundred artefacts more pertinent to the search query, we have derived two different formulations of the precision measure. In particular, based on the metrics used in (Huang, 2008), the precision measure for Google has been defined as follows:

$$P_{Google} = \frac{\sum_{i=1}^{|White|} r_i}{|White \cup Gray \cup Google|} \tag{1}$$

Where *White* and *Gray* are the lists retrieved by CAIMANS, whereas *Google* are the results retrieved by Google, and the rank r_i is a number between 0 and 1 representing the semantic similarity of the i-th artefact in *White* w.r.t. the search query.

Similarly, the precision measure for search engine in CAIMANS has been defined has follows:

$$P_{\text{CAIMANS}} = \frac{\sum_{i=1}^{|White|+|Gray|} r_i}{|White \cup Gray \cup Google|} \tag{2}$$

In addition to (1), other than the ranks of the artefacts in *White*, (2) also considers the ranks of the artefacts in *Gray*. As said above, the ranks are calculated by the semantic module. The domain expert determined the true positives among the retrieved artefacts. Analogously, the recall measures for the two compared systems have been defined according to the following formulas:

$$R_{Google} = \frac{|TP_{Google}|}{|White| + |Google|} \tag{3}$$

$$R_{\text{CAIMANS}} = \frac{|TP_{White}|}{|White| + |Google|} \tag{4}$$

where TP_{Google} and TP_{White} are the true positive results extracted from Google and CAIMANS, respectively, according to the selections made by the domain expert. In order to express the accuracy of the proposed search engine through a unique measure, the F-measure was used to combine the precision and recall metrics. Starting from the definitions of precision and recall given in (1)-(4), we have derived the following formula for the F_1-measure:

$$(F_1)_{Google} = 2 * \frac{P_{Google} * R_{\text{CAIMANS}}}{P_{Google} + R_{\text{CAIMANS}}} \tag{5}$$

$$(F_1)_{\text{CAIMANS}} = 2 * \frac{P_{\text{CAIMANS}} * R_{\text{CAIMANS}}}{P_{\text{CAIMANS}} + R_{\text{CAIMANS}}} \tag{6}$$

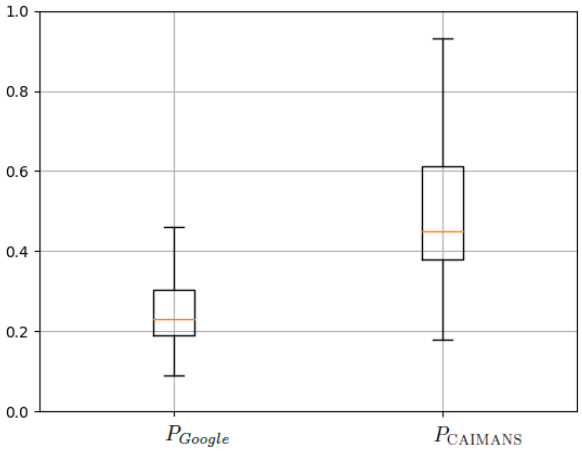


Figure 9: Precision evaluation for Google and CAIMANS.

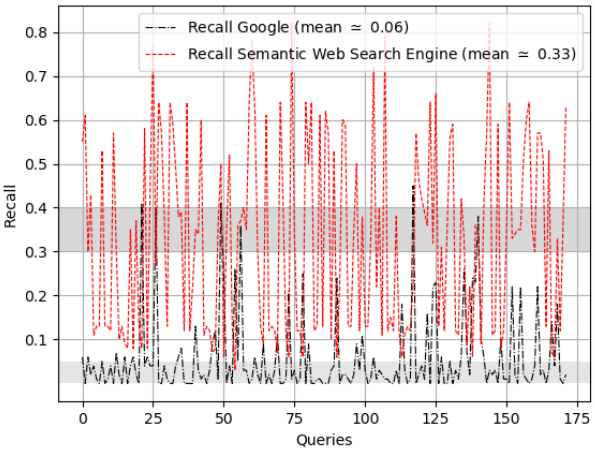


Figure 10: Recall evaluation for Google and CAIMANS.

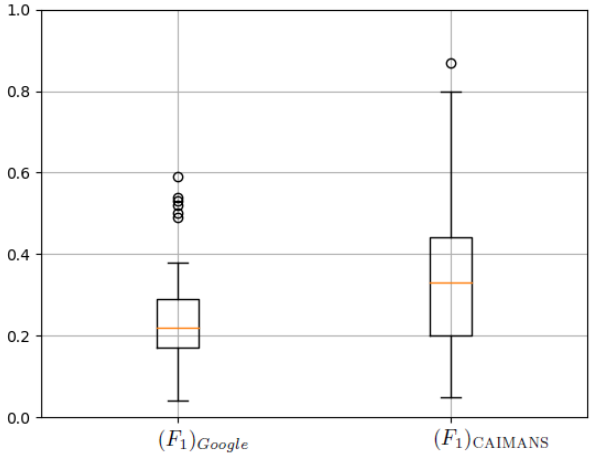


Figure 11: F-measure considered for each precision evaluation.

4.2. Semantic Web Search evaluation

In this section, we experimentally show how the cooperation between the semantic and the crawling modules improves search effectiveness with respect to traditional search engines. To this end, we compared the results achieved by CAIMANS with those achieved by Google in the e-procurement domain. In particular, in order to evaluate the correctness of results of the search sessions, we involved a user with expertise in the domain of e-procurement, focusing on the selection of findings related to the search target.

Tests were performed on a virtual machine running on a Mac with an Intel Xeon 3.20 GHz processor and 64GB of RAM. More specifically, 8GB of RAM and 250GB of local disks were dedicated to the virtual machine. The search session was accomplished using a fast connection at 340 Mbps/sec.

During the tests, we defined a single search configuration, with a unique set of keywords, URL seeds, and stopwords, aiming to carry out a peer analysis of all search sessions. As known, among millions of analysed pages, Google only shows the best 100 results, which contain all true and false positives, depending on the performed search. For this reason, and also because of the burden of the manual evaluation by the domain expert, the number of search sessions accomplished during the tests was limited to 175. Table 1 shows a part of the query strings used in our experiments. In particular, we evaluated different generated queries, such as two, three, four, and five word queries.

ID	Query String	#Words	Time of CAIMANS (s)
1	**Bandi di gara Bologna**	3	2047
2	**Onlus e gare d'appalto**	3	1922
3	**Autostrade gare appalto**	3	2279
4	**Bandi di gara Bolzano**	3	2106
5	**Bandi di gara Salerno provincia**	4	249
6	**Bandi di gara regione Sardegna**	4	2230
7	**Bandi di gara regione Calabria**	4	2238
8	**Gare d'appalto Banca d'Italia**	4	256
9	**Bandi di gara distributori automatici 2019**	5	225
10	**Bandi di gara beni culturali Puglia 2019**	5	243

ID	Google Results	TP_{Google}	CAIMANS Results	Common Results	|Black|	|Gray|	|White|	$TP_{CAIMANS}$
1	100	18	544	11	2	55	487	538
2	100	4	478	2	24	70	384	435
3	100	3	621	2	22	164	435	563
4	100	8	621	5	22	126	473	594
5	100	1	67	0	22	22	23	27
6	100	18	505	13	24	131	350	474
7	100	25	475	18	34	105	336	393
8	100	1	75	1	43	15	17	25
9	100	1	77	1	43	17	17	26
10	100	0	80	0	46	18	16	27

ID	Rate of Dynamic Pages	P_{Google}	$P_{CAIMANS}$	R_{Google}	$R_{CAIMANS}$	$(F_1)_{Google}$	$(F_1)_{CAIMANS}$
1	0,74	0,46	0,93	0,03	0,82	0,59	0,87
2	4,19	0,41	0,83	0,01	0,78	0,54	0,80
3	6,01	0,40	0,79	0,01	0,79	0,53	0,79
4	0,83	0,38	0,75	0,01	0,82	0,52	0,79
5	35,56	0,37	0,75	0,01	0,11	0,16	0,19
6	1,46	0,36	0,72	0,04	0,77	0,49	0,75
7	10,88	0,38	0,76	0,06	0,72	0,50	0,74
8	15,63	0,34	0,68	0,01	0,12	0,18	0,20
9	14,71	0,31	0,63	0,01	0,13	0,18	0,21
10	14,71	0,31	0,62	0,00	0,13	0,18	0,21

Table 1: Results achieved from query strings used in the experimental evaluation.

The first type of query involved four to five words, such as "bandi di gara beni culturali Puglia 2019" (public procurements for cultural heritage Puglia region year 2019) or "bandi di gara Salerno provincia" (public procurements in the province of Salerno). However, based on the results extracted by Google for these types of queries, the domain expert considered only a few of them as pertinent, filtering 0 calls in many cases. Viceversa, CAIMANS always retrieved a high number of valid results, even when Google's true positive results were few. The second type of query involved two to three words, such as "gare d'appalto Basilicata" (public procurements in Basilicata region) or "gare d'appalto Bologna" (public procurements in

ID	Domain	Seed URLs
1	Gare d'appalto ("Calls for tender")	https://www.gazzettaufficiale.it https://www.ooppcampania-appalti.maggiolicloud.it https://www.ansa.it
2	Robot Programmabili ("Robot Programming")	https://www.robot-advance.com https://www.wiki.ezvid.com https://www.softbankrobotics.com
3	Equipaggiamento per Hockey ("Hockey equipment supply")	https://www.decathlon.it https://www.skatepro.it https://it.hockeyoffice.com
4	Forniture per uffici ("Office supplies")	https://www.visualcapitalist.com https://www.oknoplast.it https://www.prontopro.it
5	Giochi da tavolo ("Table Games supplies")	https://www.boardgamesofferte.it https://www.boardgameitalia.it https://it.wikipedia.org
6	Cloud Provider ("Cloud providing")	https://www.zerounoweb.it https://www.meteo.it https://www.techcompany360.it

Table 2: Details of the topics and seed URLs used for the comparative evaluation on e-procurement domain.

the city of Bologna). Even though the domain expert filtered a conspicuous number of results among those returned by Google for these types of queries, with CAIMANS we could considerably increase the size of the result set.

Successively, the domain expert was asked to perform a further filtering phase, in which only the results classified in the White and the Gray lists were re-examined. Table 1 shows some results of evaluation phase, in which a set of query strings with different number of words were submitted. We can observe that the number of artefacts that Google extracted and classified as true positive (TP_{Google}) is always lower than CAIMANS (TP_{CAIMANS}). We can notice that even when Google did not find useful results, CAIMANS often extracted several pages related to the search target.

The semantic module plays a fundamental role within CAIMANS, since it discards all uninteresting results and is able to identify pages related to the search criteria, increasing the quality of extracted pages.

Figure 9 shows the precision values computed for CAIMANS, according to the formulas (1) and (2). In particular, the average value is higher when considering all the true positives of the White and the Gray lists. Moreover, the second filtering activity accomplished by the domain expert has further refined search results. The achieved precision values show that the CAIMANS provides a high number of relevant results in the focus centred search.

By using the formulas (3) and (4), it was possible to define the values of the recall, i.e. the probability that a pertinent artefact is retrieved in the focused centred search. Figure 10 compares the recall achieved by CAIMANS to that of Google. The results show that the average recall of CAIMANS is higher than the one achieved by Google. However, when generic strings were used, such as "bandi di gara" (public procurements) or its synonym "gare d'appalto" (calls for tender), the values of the recall for the two compared systems would turn out to be similar.

The main goal of this evaluation was to show the effectiveness of CAIMANS for focused search by considering different types of search operations. Although time performances of CAIMANS are quite good (see Table 1), we cannot compare them with Google, since the latter is able to browse a large part of the web in few seconds by exploiting efficient algorithms and scalable architectures. For this reason, we have chosen to design CAIMANS as a batch system able to explore the web through multiple scheduled searches. The experimental results show that the performances are quite similar for queries with a different number of words.

4.3. Comparative evaluation

In this section, we show the result of a comparative evaluation among CAIMANS and similar systems, on multiple domains. Since to the best of our knowledge in the literature there does not exist any system similar to CAIMANS, we performed such comparison on its main module, i.e., the crawler. To this end, we selected two focused crawlers, one based on Breadth-First Search (BFS) and the other on Depth First Search (DFS)[12], and plugged them in turn within the CAIMANS's architecture. A BFS crawler relies on the Breadth-First Search of a tree or chart (Bundy & Wallen, 1984). The exploration starts with the seed URLs associated to the current domain, and for each of them, the crawler saves all the links whose depth is one more than the analysed URL. After exploring all the URLs in one level, the crawler scans the pages at the next level by exploiting the same strategy. On the contrary, a DFS crawler relies on the Depth First Search of a tree or graph (Tarjan, 1972). The exploration starts with the seed URLs, and for each of them, it performs a deep scan until all URLs on that path are retrieved. Then, it goes back to scan other branches of the tree.

All the three configurations of CAIMANS with the different compared crawlers were run on the same application domains and, for each of them, we configured the semantic module to evaluate the resulting pages with Cosine Similarity, Dice Similarity (Son & Kim, 2017), and Jaccard Similarity (Niwattanakul et al., 2013) metrics.

Table 2 shows the domains and the corresponding seed URLs used in our evaluation. In particular, 6 domain were divided into two categories based on their seed URLs: the first category containing domains in which the seed URLs are closely related to the target topic: "Gare d'appalto" ("Calls for tender"), "Robot programmabili" ("Robot Programming"), and "Equipaggiamento per Hockey" ("Hockey equipment supply"), whereas the second one containing domains not directly related to the target topic: "Forniture per uffici" ("Office supplies"), "Giochi da tavolo" ("Table games supplies"), and "Cloud Provider" ("Cloud providing"). Notice that, all the tests with the three system configurations have been accomplished by considering common keywords and stopwords for each target topic.

Table 3 shows the results of the comparative evaluation on selected domains, highlighting the number of web pages retrieved and the results of the semantic module for each execution. Each crawler was run assuming that the maximum number of pages to browse was 1000. However, if all URLs within the queue were crawled, i.e., if the URL queue was empty during execution, the crawler would stop its crawling operation.

The results show that the number of web pages extracted by CAIMANS is greater than those extracted by the configuration with the DFS and BFS crawlers. This is probably due to the fact that such crawlers mainly focus their exploration on the subdomains of the seed URLs (Table 2). Viceversa, CAIMANS is able to prioritise searches in web domains beyond the subdomains of the seed URLs by exploiting a search strategy in which the order of the links within the URL queue changes continuously (Section 3.1.1).

Concerning the comparative evaluation of similarity metrics, results show that the number of pages of interest evaluated with Cosine Similarity is always greater than the one achieved with other similarity metrics on CAIMANS and BFS crawlers. Moreover, it is important to notice that several results classified as not relevant (i.e., black list) from the semantic modules with Dice and Jaccard metrics have been evaluated as uncertain relevant results (i.e., grey list) with the Cosine Similarity metrics. For this reason, these results have been individually assessed, and all of them have been classified as artefacts of interest.

Although these results show the effectiveness of the Cosine Similarity, results in Table 3 show that there are some exceptions for DFS crawler in which the number of pages of interest evaluated with Dice and Jaccard similarity metrics is greater than the one achieved with the Cosine Similarity. In particular, the semantic module configured with the Jaccard Similirarity for the DFS crawler outperforms the results of other modules for the topics "Robot Programming" and "Table Games supplies" (i.e., Topic 2 and 5). However, several results in the white list have been included in the grey lists of the other two semantic modules configured with Dice and Cosine Similarity, respectively.

[12]https://github.com/ethanZHY/Crawler_BFS_DFS_Python3

Topic	CAIMANS		Cosine Similarity			Dice Similarity			Jaccard Similarity		
	Results	Time (s)	|Black|	|Gray|	|White|	|Black|	|Gray|	|White|	|Black|	|Gray|	|White|
1	882	1541	377	250	255	494	265	123	597	198	87
2	241	1289	101	119	21	152	76	13	150	82	9
3	134	1276	59	47	28	76	37	21	85	45	4
4	859	1541	298	303	258	421	234	204	476	214	169
5	47	97	9	11	27	18	3	26	11	27	9
6	241	261	51	68	122	130	53	78	72	77	112

Topic	DFS		Cosine Similarity			Dice Similarity			Jaccard Similarity		
	Results	Time (s)	|Black|	|Gray|	|White|	|Black|	|Gray|	|White|	|Black|	|Gray|	|White|
1	31	10520	8	17	6	12	17	2	16	11	4
2	68	5546	25	38	5	37	25	3	42	7	19
3	22	8854	7	11	4	15	6	1	10	8	4
4	33	11201	22	8	3	17	13	3	25	6	2
5	37	9476	20	9	8	22	10	5	14	13	10
6	60	2272	34	18	8	31	21	8	17	37	6

Topic	BFS		Cosine Similarity			Dice Similarity			Jaccard Similarity		
	Results	Time (s)	|Black|	|Gray|	|White|	|Black|	|Gray|	|White|	|Black|	|Gray|	|White|
1	20	135	20	0	0	17	3	0	19	1	0
2	4	4437	3	0	1	4	0	0	3	0	1
3	18	1404	12	1	5	10	5	3	12	4	2
4	149	2625	39	98	12	72	69	8	71	73	5
5	29	240	10	13	6	16	7	6	15	9	5
6	9	186	6	3	0	8	1	0	9	0	0

Table 3: Results of the comparative evaluation on general topics.

5. Conclusion

In this paper, we have proposed CAIMANS, an intelligent system aiming to effectively support companies in the process of selecting artefacts of interest from the web, verifying how they match company's backgrounds and expertise stored in its data and knowledge sources. Thus, CAIMANS relies on an advanced crawler to search artefacts of interest from the web. An extended validation has been carried out in cooperation with industrial stakeholders on real size use cases from the e-procurement domain, proving the capabilities of CAIMANS to improve the effectiveness of the calls for tender management process with respect to traditional practices relying on current search engines and human expertise. An additional comparative evaluation has been accomplished to compare the crawler of CAIMANS with to well-known focused crawlers, namely BFS and DFS, on several other domains other than e-procurement. The crawler of CAIMANS outperformed BFS and DFS on 6 domains.

In the future, we would like to further develop the company's data and knowledge source component, by modelling additional information concerning the company's resources, including internal skills, partnerships, and policies. This would allow us to conceive more an advanced module for contrasting the results of the crawling phase for the company's skills and capabilities. Moreover, since CAIMANS relies on general methodologies, it is not bounded to a single application domain. Thus, in the future, it is possible to consider the application of different public procurement contexts where the search for calls exploits non-structured artefacts. Moreover, we would like to further extend the CAIMANS system by evaluating the possibility of embedding more recent query expansion techniques within its crawling components.

Acknowledgment

This work has been supported by the research project named "PROBIM" for the regional research centre named as CeRICT scrl.

References

Acharya, S., & Parija, S. (2010). The process of information extraction through natural language processing. *International Journal of Logic and Computation (IJLP)*, *1*, 40–51.

Al-Nabki, M. W., Fidalgo, E., Alegre, E., & Fernández-Robles, L. (2019). Torank: Identifying the most influential suspicious domains in the tor network. *Expert Systems with Applications*, *123*, 212–226.

Arumawadu, H. I., Rathnayaka, R., & Illangarathne, S. (2015). *K-Means Clustering For Segment Web Search Results*. Kambohwell Publisher Enterprises.

Balbi, S., Misuraca, M., & Scepi, G. (2018). Combining different evaluation systems on social media for measuring user satisfaction. *Information Processing & Management*, *54*, 674–685.

Bidoki, A. M. Z., & Yazdani, N. (2008). Distancerank: An intelligent ranking algorithm for web pages. *Information Processing & Management*, *44*, 877–892.

Bifulco, I., & Cirillo, S. (2018). Discovery multiple data structures in big data through global optimization and clustering methods. In *Proceedinfs of the 22nd International Conference Information Visualisation (IV)* (pp. 117–121).

Borg, I., & Groenen, P. (2003). Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement*, *40*, 277–280.

Broder, A. Z., Glassman, S. C., Manasse, M. S., & Zweig, G. (1997). Syntactic clustering of the web. *Computer networks and ISDN systems*, *29*, 1157–1166.

Bundy, A., & Wallen, L. (1984). Breadth-first search. In *Catalogue of artificial intelligence tools* (pp. 13–13).

Cambazoglu, B. B., Karaca, E., Kucukyilmaz, T., Turk, A., & Aykanat, C. (2007). Architecture of a grid-enabled web search engine. *Information Processing & Management*, *43*, 609–623.

Carloni, E. (2005). Codice dell'amministrazione digitale commento al d. lgs (in italian). `http://www.amministrazioneincammino.luiss.it/2005/11/16/enrico-carloni-a-cura-di-codice-dell%E2%80%99amministrazione-digitale-commento-al-d-lgs-7-marzo-2005-n-82-maggioli-editore-rimini-2005/`. Accessed: 2021-01-07.

Carpineto, C., & Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys*, *44*, 1–50.

Caruccio, L., Deufemia, V., & Polese, G. (2017). Learning effective query management strategies from big data. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 643–648). IEEE.

Cavaness, C. (2006). *Quartz Job Scheduling Framework: Building Open Source Enterprise Applications*. Pearson Education.

Chakrabarti, S., Van den Berg, M., & Dom, B. (1999). Focused crawling: a new approach to topic-specific web resource discovery. *Computer networks*, *31*, 1623–1640.

Choy, D., Brown, A., Gur-Esh, E., McVeigh, R., & Muller, F. (2010). "content management interoperability services (cmis), version 1.0". OASIS Standard, available on line at `http://docs.oasis-open.org/cmis/CMIS/v1.0/cmis-spec-v1.0.html`. Accessed: 2021-01-07.

De Souza, C. R. (2012). A tutorial on principal component analysis with the accord. net framework. *arXiv preprint arXiv:1210.7463*, .

Dhingra, V., & Bhatia, K. K. (2015). Semcrawl: framework for crawling ontology annotated web documents for intelligent information retrieval. In *Intelligent Distributed Computing* (pp. 213–223).

Dong, H., & Hussain, F. K. (2010). Focused crawling for automatic service discovery, annotation, and classification in industrial digital ecosystems. *IEEE Transactions on Industrial Electronics*, *58*, 2106–2116.

Du, Y., Liu, W., Lv, X., & Peng, G. (2015). An improved focused crawler based on semantic similarity vector space model. *Applied Soft Computing*, *36*, 392–407.

Foucault, N., Adda, G., & Rosset, S. (2011). Language modeling for document selection in question answering. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011* (pp. 716–720).

Ghorab, M. R., Zhou, D., O'connor, A., & Wade, V. (2013). Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, *23*, 381–443.

Goel, S., Kumar, R., Kumar, M., & Chopra, V. (2019). An efficient page ranking approach based on vector norms using snorm (p) algorithm. *Information Processing & Management*, *56*, 1053–1066.

Gove, R. (2015). Using the elbow method to determine the optimal number of clusters for k-means clustering. `https://bl.ocks.org/rpgove/0060ff3b656618e9136b`. Accessed: 2021-01-07.

Grosman, J. S., Furtado, P. H., Rodrigues, A. M., Schardong, G. G., Barbosa, S. D., & Lopes, H. C. (2020). Eras: Improving the quality control in the annotation process for natural language processing tasks. *Information Systems*, (p. 101553).

Hernández, I., Rivero, C. R., & Ruiz, D. (2019). Deep web crawling: a survey. *World Wide Web*, *22*, 1577–1610.

Heydon, A., & Najork, M. (1999). Mercator: A scalable, extensible web crawler. *World Wide Web*, *2*, 219–229.

Hilbert, M., & López, P. (). The world's technological capacity to store, communicate, and compute information. *Science*, *332*, 60–65.

Hu, G., Zhou, S., Guan, J., & Hu, X. (2008). Towards effective document clustering: A constrained k-means based approach. *Information Processing & Management*, *44*, 1397–1409.

Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC)* (pp. 9–56). volume 4.

Jain, N., Mangal, M. P., & Bhansali, A. (2013). An approach to build a web crawler using clustering based k-means algorithm. *Journal of Global Research in Computer Science*, *4*, 14–22.

Jones, M., Campbell, B., & Mortimore, C. (2015). Json web token (jwt), profile for oauth 2.0 client authentication and authorization grants. `https://tools.ietf.org/html/rfc7523`. Accessed: 2021-01-07.

Kim, H., Kim, H. K., & Cho, S. (2020). Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling. *Expert Systems with Applications*, *150*, 113288.

Kumar, M., Bhatia, R., & Rattan, D. (2017). A survey of web crawlers for information retrieval. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*, e1218.

Kumar, M., & Vig, R. (2013). Focused crawling based upon tf-idf semantics and hub score learning. *Journal of Emerging technologies in web intelligence*, *5*, 70–77.

Lakshmi, R., & Baskar, S. (2019). Novel term weighting schemes for document representation based on ranking of terms and fuzzy logic with semantic relationship of terms. *Expert Systems with Applications*, *137*, 493–503.

Langari, R. K., Sardar, S., Mousavi, S. A. A., & Radfar, R. (2020). Combined fuzzy clustering and firefly algorithm for privacy preserving in social networks. *Expert Systems with Applications*, *141*, 112968.

Lempel, R., & Moran, S. (2000). The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Computer Networks*, *33*, 387–401.

Lensen, A., Xue, B., & Zhang, M. (2017). Using particle swarm optimisation and the silhouette metric to estimate the number of clusters, select features, and perform clustering. In *Proceedings of the European Conference on the Applications of Evolutionary Computation* (pp. 538–554).

Liu, C.-L., Hsaio, W.-H., Lee, C.-H., & Chen, C.-H. (2013). Clustering tagged documents with labeled and unlabeled documents. *Information Processing & Management*, *49*, 596–606.

Lozano, S., & Calzada-Infante, L. (2019). Efficiency ranking using dominance network and multiobjective optimization indexes. *Expert Systems with Applications*, *126*, 83–91.

Manku, G. S., Jain, A., & Das Sarma, A. (2007). Detecting near-duplicates for web crawling. In *Proceedings of the 16th international conference on World Wide Web* (pp. 141–150).

Manyika, J. (2011). Big data: The next frontier for innovation, competition, and productivity. *Technology and Innovation Big data The next frontier for innovation*, .

McDaniel, M., & Storey, V. C. (2019). Evaluating domain ontologies: Clarification, classification, and challenges. *ACM Computing Surveys*, *52*.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists* (pp. 380–384). volume 1.

Paivarinta, T., & Munkvold, B. E. (2005). Enterprise content management: an integrated perspective on information management. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, (pp. 96–96).

Pandey, S., & Olston, C. (2005). User-centric web crawling. In *Proceedings of the 14th international conference on World Wide Web* (pp. 401–411).

Pinnamaneni, L. C. (2013). Intelligent web crawling using semantic signatures. `https://researchrepository.wvu.edu/etd/4989`. Accessed: 2021-01-07.

Premlatha, K., & Geetha, T. (2011). Focused crawling for educational materials from the web. *International Journal of Computer Science & Informatics*, *1*, 26–29.

Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (pp. 133–142). volume 242.

Schonfeld, U., Bar-Yossef, Z., & Keidar, I. (2006). Do not crawl in the dust: different urls with similar text. In *Proceedings of the 15th international conference on World Wide Web* (pp. 1015–1016).

Son, J., & Kim, S. B. (2017). Content-based filtering for recommendation systems using multiattribute networks. *Expert Systems with Applications*, *89*, 404–412.

Sun, Z., Lim, E.-P., Chang, K., Ong, T.-K., & Gunaratna, R. K. (2005). Event-driven document selection for terrorism information extraction. In *Proceedings of the International conference on intelligence and security informatics* (pp. 37–48).

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, *1*, 146–160.

Vassilvitskii, S., & Arthur, D. (2006). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027–1035).

Vattani, A. (2011). K-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, *45*, 596–616.

Vidal, M. L., da Silva, A. S., de Moura, E. S., & Cavalcanti, J. (2006). Structure-driven crawler generation by example. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 292–299).

Yuvarani, M., c. s. n. Iyengar, N., & Kannan, A. (2006). Lscrawler: A framework for an enhanced focused web crawler based on link semantics. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'06)* (pp. 794–800).