

Continuous Multi-Source Information Gathering and Classification

R. Steinberger, B. Pouliquen, S. Scheer, and A. Ribeiro
European Commission, Joint Research Centre
Institute for the Protection and Security of the Citizen
I-21020 Ispra (VA), Italy

E-mail: {Ralf.Steinberger, Bruno.Pouliquen, Stefan.Scheer, Antonio.Ribeiro}@jrc.it

Abstract

This paper describes a fully functional prototype of a multi-component system that allows users to retrieve, store and search documents from a variety of publicly available information sources, in a variety of languages, on any subject domain users may be interested in.

The system integrates a crawler, which selects and downloads potentially relevant documents, a cleaning tool which removes irrelevant information from the retrieved documents (e.g. advertisements), a document filter, a document classifier, and several other tools which extract meta-information from texts such as titles and keywords. Unlike ad-hoc search engines, the system satisfies long-term information needs of users since it continuously collects documents they may be interested in. The functionality of the system has been evaluated by comparing it with a traditional manual press clipping service, and it proved to give good results.

1 Introduction

The abundance of publicly available information sources is creating new needs for intelligent tools to manage large amounts of documents. Internet search engines are getting better at helping users to satisfy *ad-hoc* information needs, but for many private persons, companies, government sectors, and particularly for intelligence departments, *ad-hoc* information retrieval tools are not enough. They do not usually provide classification schemes which rank and sort documents according to their contents.

In this paper we describe a system which fills this need. It searches publicly available information sources for documents about specific subjects and downloads and classifies them for later use.

The initial motivation to build such a system came from a request from the European Parliament to the European Commission's *Joint Research Centre* (JRC) to support the fight against *Internet abuse*, including cases of hacking, credit card fraud and paedophilia. It was decided to start by making an inventory of publicly available information on all kinds of Internet abuse in order to check whether the known cases were only the so-called 'tip of the iceberg'. The project started in early 2000 and was called OSILIA ('Open Source Information Library on Internet Abuse'). It produced a fully functional prototype and a list of concrete ideas on how to extend the functionality of the system and to generalise its usability to further domains of interest. Since the end of OSILIA, the tools have been improved and parameterised so that the system can be applied to further domains.

This paper is structured as follows: the next section gives a general overview on the methodology used to collect and select the texts. Section 3 describes each of the system components in more detail. Section 4 gives some results and provides some evaluation. Finally, we present some conclusions and future work in section 5.

2 Methodology

We started this project by collecting information on the different kinds of Internet abuse to evaluate the scale of the problem, should the cases made public be only the ‘tip of the iceberg’. Thus, we monitored publicly available news sources in two languages – English and German – so as to gather as much information as possible. The sources monitored were chosen to be twenty online newspapers and meta-news sites (sites which contain news collected from other newspaper sites) in both languages. The plan was to extend the method to further languages, should it be able to collect much useful information.

We developed tools to search and retrieve online information automatically. One of the main priorities of the project was on the storage of retrieved information. A database of documents was needed so that the documents collected could then be queried. Our experience showed that many online documents are either removed or made unavailable later so that it is important to store them temporarily. In contrast with search engines that satisfy more *ad-hoc* user needs, our system cleans the retrieved web pages, filters, classifies and stores them.

In parallel, the JRC’s Publications Office was given the task to provide a manual newspaper clipping service to allow comparing and evaluating the results of our system (see section 4).

3 Description of the System

In this section, we start by giving a general description of the system and then we describe each of the integrated tools in more detail.

3.1 Overview

An intelligent agent was launched daily to crawl the Internet. It was driven by instructions given in two types of parameter files so as to guide it through specific websites:

- one language-specific parameter file, which contains the set of search words, their weights according to the subject types and the threshold values which make a document relevant to be stored. E.g. the search pattern `paedoph*` was assigned weight 5 for the subject type ‘Crime using Internet’, but weight 0 for ‘Computer-based Fraud’; and,
- a set of source-specific parameter files, which includes information regarding the web search such as the site address, the search depth and what sections should not be crawled.

The retrieved HTML files that satisfied the requirements were then retrieved and cleaned since most of them contained textual information which was not relevant, like advertisements or links to other sections of the website. This cleaning step was crucial because many of the search words could be found in the HTML file, but outside the newspaper article itself. So, this module needed to be developed with special care in order to avoid text *noise*. It also identifies the main parts of the document like the document title, the publication date and the name of the author, where available.

A language recognition tool was used to identify the language of the few retrieved documents written in languages other than the ones expected.

Next, we used a lemmatiser to replace all inflected words by their base form. This proved to be convenient in order to improve the identification of keywords and also the filtering of relevant texts. For this task, we used *IntelliScope Search Enhancer* (version 2.0), a lemmatiser distributed by the former *Lernout & Hauspie*. This lemmatiser is also able to normalise regional spelling variations, date and currency expressions. It was selected because, in contrast with other products, it is available for a wide variety of languages.

After the cleaning stage, the texts were filtered again and they were classified into one or more user-defined classes.

We also used two other tools, which will be discussed in detail next:

- a statistical keyword identification tool which identifies the most pertinent words for each document; and,
- a document storage and display tool which lets users query the set of stored documents via the internet and display their relatedness.

The keywords give the users a general idea of the document contents in order to help them decide quickly whether the document is relevant for their interests.

3.2 Internet Crawler

The crawler is based on a PERL module (`LWP::UserAgent`) which provides a simple programming interface for automatic web document retrieval. It is launched individually for each web source and follows the instructions found in the source-specific parameter files.

For each web site to be crawled, there is one source parameter file which contains:

- the starting web site addresses;
- the search depth;
- a list of directories, file types and file names not to be searched and downloaded;
- a Boolean search term query which is used by the crawler to decide which pages are potentially relevant;
- information on where to find a text-only (or 'printer-friendly') version of the page on the site (if such a version exists);
- information on where to find the title and author of the article in the HTML page;
- roaming information to decide whether the crawler should follow links to external web sites, within the same domain, or even only inside the same directory; and,
- information on where to store the files.

The parameter files also contain performance-related information that instructs the crawler on how to proceed. This includes instructions on whether to proceed depth- or breadth-first, on the file types which should be downloaded (text, audio, video), on the maximum number of files to retrieve, on the maximum size of the files to be retrieved, and on the number of connection trials in case it is not possible to establish the connection or if it times out.

For each language, there is one parameter file which contains the search query and a set of domain-specific stop words. Words in these stop word lists will never be identified as keywords for a text. In the case of 'Internet Abuse', for instance, words such as 'Internet', 'web', or 'http' are useful to determine that the text is about the Internet, but they fail to be useful as keywords since all documents in this interest profile are about the Internet.

There is also a default parameter file which provides default values for those settings which are not mentioned in the source-specific parameter files. This file speeds up the process of creating a new source-specific parameter file. For instance, we realised that breadth-first searches are usually better than depth-first searches.

Each time the crawler finds a page satisfying the criteria given in both the source-specific and language-specific parameter files, it downloads the web page and follows the links found on that page. In this way, the crawler can search targeted web sites for specific pages that satisfy user-defined conditions.

3.3 Cleaning of Web Pages

After downloading potentially relevant articles, the source-specific parameter file provides instructions on how to identify the title, the author and the publication date of the articles. The publication date is particularly important because the crawler looks for new articles every day, but some can be older because they might have been found in the newspaper's archives. Publication and downloading dates are therefore kept in the database separately.

Experience showed that many pages were downloaded just because some or even all of the search words were found on the web page outside the newspaper article, i.e. in advertisement banners, menu bars, headers, footers, side bars, lists of other news articles of the day or of other related articles (see Figure 1).



Figure 1. A news page found for the query 'Internet' and 'fraud'. The article is not relevant for this query because 'Internet' appears only in the side bar instead of in the article.

We believe that the rather high rate of wrong hits due to surrounding irrelevant text is likely to be specific to online newspapers. Since the subject types of our queries contained words which appear often in headers and footers, such as 'internet', 'web', 'site', or 'computer', the rate of wrong hits was high. Thus, the cleaning process was absolutely necessary.

3.4 Language Recognition and Lemmatisation

The language of specific online newspapers is known, but during this project we also searched meta-news sites like Quicklinks (<http://www.qlinks.net>) or Paperball (<http://www.paperball.de>). These sites list titles of articles from a large number of different newspapers and provide links to the original locations. As some of these articles were written in languages other than the ones expected, and the crawler downloaded some of them, it was necessary to submit the filtered articles to a language recognition tool in order to exclude them. The tool we used is a statistical bigram-based language recogniser developed at the JRC [1]. Knowing the text language was necessary as both the lemmatiser and the keyword identification tool have settings which depend on it.

Next, texts were lemmatised, i.e. all words were replaced by their base form (their lemmas). This normalisation was required to improve the performance of the keyword identification tool and it also helped to improve the subsequent filtering and classification results.

3.5 Filtering and Classification

As some documents were downloaded because search words appeared in the web page, but outside the newspaper article itself, it was necessary to filter them a second time after the cleaning process. The same Boolean search query used by the crawler was applied once more, but it proved to be better to assign weights to words or to conditions in order to have a finer filtering of the documents retrieved.

The classification tool also read instructions from the language-specific parameter file on how to classify a given document into one or more of the user-defined classes. In OSILIA, four main classes were distinguished:

1. Attacks on Computers: e.g. viruses, Trojan horses;
2. Attacks on Internet: e.g. worms, denial-of-service attacks, hacking of web sites;
3. Computer-based Fraud: e.g. cracking, credit card stripping, attacks on financial encryption; and,
4. Crime using Internet: e.g. paedophilia, terrorism, instructions for bomb-making.

In OSILIA, classification was done by adding the weight of words occurring in the text to the score of a class. The tables containing the list of words and their numerical weights were defined by the user and stored in the language-specific parameter file. For each class, the user also had to decide on the threshold to be reached for a document to be assigned to that particular class. Documents could be assigned to more than one class if they reached the minimum threshold of several classes. This weighting mechanism acted as an additional relevance-ranking filter because documents which did not get assigned to any class were clearly less relevant to the subject type.

The classification information was stored in the database, together with the title, the language, the keywords of the text, and other text-specific information.

3.6 Keyword Identification

The keyword identification tool chooses a limited set of particularly relevant words from each text for storage in the database. It is a purely statistical tool that compares a lemma frequency table for a given text with a lemma frequency table of a reference corpus in order to identify the most salient words of the text, using a choice of standard statistical techniques.

In other words, if a certain lemma occurs significantly more often in a given text than it occurs, on average, in a large selection of ‘normal’ texts (the reference corpus), then this word is identified as a keyword. For the assignment of keywords to general newspaper articles, the best reference corpus is usually a balanced corpus such as the *British National Corpus* [2] or a collection of several years of newspaper articles, as they should have information on the distribution of words from different subject domains. Keyword lists of texts on ‘Internet Abuse’ will quite correctly contain words such as ‘Internet’, ‘web’, ‘http’, ‘page’. However, these words are redundant within this particular collection of texts because they were a basic condition for the retrieval of those documents. For this reason, it is appropriate to add these sublanguage-specific content words to the stop word list in the language-specific parameter file so that the keyword assignment tool does not identify them as keywords.

The statistical tool currently uses two alternative tests to identify the most relevant words of a text: the *chi-square* and the *log-likelihood* tests. We mainly use the latter algorithm because it works better for low-frequency words [3]. The tool not only produces a list of the most significant words of a text, but it also gives a score on their importance as content descriptors for this document. We refer to this indicator as the *keyness* of the keyword. Table 1 shows a sample list of keywords identified automatically for the document you are currently reading.

| Keywords | Keyness | Keywords | Keyness | Keywords | Keyness |
|-----------------|----------------|-----------------|----------------|-----------------------|----------------|
| document | 710.8 | language | 179.3 | gist | 109.1 |
| internet | 404.4 | article | 176.7 | jrc | 100.9 |
| keyword | 402.8 | file | 173.3 | eurovoc | 99.0 |
| text | 282.7 | user | 168.1 | joint_research_centre | 99.0 |
| search | 242.0 | parameter | 156.9 | lemmatiser | 99.0 |
| tool | 232.3 | specific | 124.5 | steinberger | 92.4 |
| download | 223.8 | domain | 124.5 | retrieve | 91.9 |
| web | 205.9 | html | 118.8 | classification | 85.6 |
| crawler | 204.4 | osilia | 118.8 | meta | 83.9 |
| information | 201.3 | source | 112.3 | hagman | 81.7 |
| http | 198.0 | newspaper | 111.8 | assign | 80.4 |
| word | 187.8 | identify | 111.6 | | |

Table 1. Automatically identified keywords for this document and their keyness scores [4].

The tool produces accurate lists of keywords. Not all suggested keywords are highly meaningful, but they normally provide users with a rather good idea of the document contents. Figure 2 shows how these keywords can be used to browse the document collection. The advantage of searching for documents to which a certain keyword has been assigned as opposed to searching for documents simply containing this word is that the user is guaranteed that this word was not just mentioned on the side, but that it is of a certain importance in the documents. The tool can be applied to any language – all it needs is a word frequency list of a reference corpus and, preferably, a lemmatiser or stemmer for the new language.

3.7 Storage, Visualisation and Searching

A few weeks after the crawler had been set up to run, the collection consisted of a rather large number of documents. It was therefore useful to provide a search interface that allows the users to retrieve specific documents or to browse the collection using a variety of criteria.

Such criteria can be, for instance:

- author name;
- document title;
- publisher name;
- document language;
- keywords;
- subject classification;
- publication date;
- downloading date.

Similar requirements have been found in other projects using the *Generic Information Server Toolkit* (GIST) software¹[5]. Therefore, we used it in this application, as well. GIST provides facilities for the construction of information object repositories and for their management via the web. In order to build a GIST-based system, one starts by defining a data model, i.e. a description of the information objects to be managed. GIST uses this model to guide its behaviour when adding and manipulating the repository contents. Once the data model is defined, objects can be inserted into the repository.

The screenshot shows the OSILIA web interface. At the top left is the logo 'OSILIA' and 'home'. At the top right is 'browse categories'. Below the logo is a navigation menu with links: [register], [login], [search], [what's new], [calendar], [view types], [discussions], and [small text]. In the main content area, there is a button that says '[list 1418 documents now]'. Below this, there are two sections: 'document type' and 'subject type'. The 'document type' section lists 'Web Article, (1416)' and 'Other, (2)'. The 'subject type' section lists 'Other, (2)', 'crime using internet, (130)', 'computer based fraud, (459)', 'attacks on Internet, (327)', and 'attacks on computers, (500)'. Below these is a section for 'top 500 keywords' listing various terms and their counts, such as 'hacker, (414)', 'virus, (291)', 'site, (280)', 'microsoft, (247)', 'fbi, (212)', 'e-mail, (207)', 'bug, (182)', 'love, (173)', 'email, (167)', 'software, (160)', 'crime, (140)', 'security, (139)', 'law, (120)', 'network, (108)', 'download, (106)', 'user, (104)', and 'cybercrime, (99)'.

Figure 2. GIST search interface: *browse by subject types and keywords* function (<http://osilia.jrc.it>).

¹ GIST is a toolkit developed at the JRC for the rapid development of interactive web-based information servers. It removes the technical barriers traditionally associated with creating interactive web sites. It has been specifically designed to allow user communities to share information and communicate more effectively without the need for a full-time technical web master.

A data model for this methodology includes a description of ‘document’-type objects based on the characteristics identified above. Each document (file) is a separate object with a title, description, publisher and other document-specific information.

Since we do not publish the retrieved articles on the Internet, we did not feel pushed to use an established metadata element set to describe the document contents, like the Dublin Core [6][7]. However, we do analyse texts in order to extract and produce an equivalent to all Dublin Core elements.

The Dublin Core elements we are currently extracting from the source, directly or indirectly, are: ‘Title’, ‘Publisher’, ‘Date’, ‘Format’ (e.g. MIME), ‘Source’ (e.g. newspaper), ‘Identifier’ (e.g. URL) and ‘Type’ (online newspaper article; Dublin Core specifications suggest ‘text’ [8], but this is too general for our purposes).

Furthermore, we are able to identify more elements, and, consequently, we enrich the resource with further meta-data:

- Language: we use the language recognition tool to identify the language, and we encode it with the two-letter code standard ISO 639²;
- Subject: in addition to the subject types, a set of keywords is used to describe each of the documents;
- Description: this can be an abstract or a table of contents;
- Relation: we identify related documents in our document database, or user-defined classes;
- Coverage: we can identify geographical references; and,
- Creator: we identify the author name or the name of the news agency.

Figure 3. GIST search interface.

² <http://lcweb.loc.gov/standards/iso639-2/langcodes.html>.

GIST supports the addition of information objects via HTML forms and XML files. XML files are more suitable for automated import, so an XML description of each document must be generated. The user interface for a GIST repository (Figure 2 and Figure 3) is produced using template files containing HTML text and embedded commands for formatting and control.

4 Evaluation

As mentioned earlier in section 2, the performance of the system was judged by comparing the automatically gathered newspaper articles with a manual collection of press clippings compiled in parallel by the Public Relations department of the JRC. The evaluation was carried out by a user with a background in journalism.

The Public Relations department collected about 300 documents and our system collected more than 1400 documents on the various subject types:

| Subject Type | Number of documents |
|----------------------|---------------------|
| Attacks on Computers | 500 |
| Attacks on Internet | 327 |
| Computer-based Fraud | 459 |
| Crime using Internet | 130 |
| Total | 1416 |

Table 2. Number of documents collected for each subject type.

The evaluator was very satisfied with the results and confirmed that all major issues that had been collected manually were also found by the automatic procedure. This means that our system was able to provide a high recall value. As for precision, the evaluator confirmed that 95% of the downloaded documents covered the subject *Internet abuse* either directly or indirectly. The remaining 5% were presumably downloaded because the search words were still found in advertising links, in links to other articles of the day, or in other texts not pertaining to the news article as such even though we had put some effort on the cleaning of noisy text. Another explanation is that the search words did not carry the meaning expected or were unrelated to one another. For instance, “An internet report on child abuse in local authority homes” scored both for ‘Internet’ and ‘child abuse’. However, the fact that the report was published on the Internet was accidental with respect to the child abuse case. Thus, the article ended up being irrelevant for our purposes.

Only about 20% of the downloaded articles were considered to be really innovative or important, meaning that they actually contained important *new* information. This relatively low number has to do with the fact that some events, such as the striking of the *I-love-you* virus, were discussed redundantly and highly similar articles were found in many different newspapers, without providing much new information. They were also the subject of ‘update’ articles published daily or weekly, many of which contained the same information. This is to be expected for popular front-cover events, like this one, which attracted much public attention. This specific problem will probably be somewhat smaller when searching for lower-profile news stories.

While the comparison of the manual newspaper clipping proved that the system is able to produce a very good recall, i.e. existing information was found, the user group noted that not much more information on *Internet abuse* is published in the press than can already be found

in specialist publications. The hope that the system would reveal larger parts of the so-called 'tip of the iceberg' (information easily visible to everyone) was not fulfilled. The reason for this may be that companies affected by Internet attacks will do their best not to publicise the event. A look at specialist newsgroups may yield more insider information.

While the good performance of the system had a negative outcome (less information is available on this type of open sources than expected), the tool set was technically successful and can therefore be extremely useful for other subject domains and uses.

5 Conclusion

In this paper we have presented a system developed to search documents on specific subjects from publicly available information sources. It collects, stores and classifies the retrieved documents for later use. At this moment, the system is also able to handle documents in PDF, Postscript and Microsoft Word formats, by converting them to plain text.

The results have shown that 95% of the documents retrieved were relevant for the subject domain dealt with, even though only about 20% of them did actually bring previously unknown information.

A number of improvements are currently being implemented, which will enhance the functionality of the system. We plan to add email notification functionality to the database in order to inform users in regular intervals of the most recent articles that have been added to the collection. This will make the system better suited for people who need to be kept updated of recent events.

In addition to the keyword identification described above, we plan to assign controlled vocabulary keywords from the multilingual thesaurus EUROVOC [4]. Assigning descriptors from this closed list of terms should make the searching process easier because users experienced with this thesaurus know which keywords to expect and to use to search for documents. The major advantage of this added functionality would be that the keyword assignment is cross-lingual, meaning that English keywords can be displayed for a German text, and vice versa. The language-independent EUROVOC descriptors will also allow cross-language retrieval in the database because user queries formulated using the EUROVOC descriptors in one language will also yield documents that are present in the database but written in other languages.

We also plan to calculate the document similarity [9][10] between all documents in the database. In this way, when users identify a document of interest, they can ask for a list of the most similar documents to the one chosen. By linking all documents to the same multilingual EUROVOC thesaurus, it will be possible to calculate the document similarity between documents even if they are written in different languages [11].

The current classification system uses counters to compute the score for classification. This simple strategy was chosen because initially the classes were empty, i.e. no documents were present in them, and it was necessary to define the set of documents related to each class. However, once the classes are populated with a significant number of correctly classified texts, classification can be automated by calculating the similarity between any new text and the documents in each class, e.g. using a statistical method involving multidimensional analysis [9][12].

Finally, many users are interested in names of people and organisations mentioned in texts. We are planning to get a tool which recognises references to named entities in text (a named-entity recognition tool). Any meta-information extracted from the textual data can be made searchable via the database and would therefore improve the usefulness of the database containing the text collection.

References

- [1] J. Hagman, *Construction and Performance of a Language Recognizer*, Modus Operandi project – deliverable 8, Joint Research Centre Technical Note No. I.00.108, 1999.
- [2] L. Burnard (ed.), *British National Corpus – Users Reference Guide*, version 1.0, Oxford University Computing Services, Oxford, UK, 1995 (available at <http://info.ox.ac.uk/bnc/index.html>).
- [3] A. Kilgariff, Which Words are Particularly Characteristic of a Text? A Survey of Statistical Approaches, *Artificial Intelligence and the Simulation of Behaviour (AISB) Workshop on Language Engineering for Document Analysis and Recognition*, Brighton, UK, 33–40, April 1996.
- [4] R. Steinberger, Cross-lingual Keyword Assignment, *XVII Conference of the Spanish Society for Natural Language Processing (SEPLN'2001)*, Jaén, Spain, 273–280, September 2001.
- [5] P. Henshaw & P. Shiels, *GIST (Generic Information Server Toolkit) Overview*, Joint Research Centre Technical Note No. I.00.137, 2000.
- [6] Dublin Core Meta Data Initiative, *Dublin Core Metadata Element Set, Version 1.1: Reference Description*, 1999 (available at <http://dublincore.org/documents/dces/>).
- [7] D. Hillmann, *Using Dublin Core*, 2001 (available at <http://dublincore.org/documents/usageguide/>).
- [8] Dublin Core Meta Data Initiative, *DCMI Type Vocabulary*, 2000 (available at <http://dublincore.org/documents/dcmi-type-vocabulary/>).
- [9] J. Hagman, *An Implemented Cluster Analyzer for Documents and their Indexing Terms*, Modus Operandi project – deliverable 12A, Joint Research Centre Technical Note No. I.00.106, 1999.
- [10] G. Salton & M. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, USA, 1983.
- [11] R. Steinberger, B. Pouliquen & J. Hagman, Cross-lingual Document Similarity Calculation Using the Multilingual Thesaurus Eurovoc, *Third International Conference on Intelligent Text Processing and Computational Linguistics*, A. Gelbukh (ed.), Lecture Notes in Computer Science series, Vol. 2276, Springer, Berlin, Germany, 415–424, 2002.
- [12] F. Murtagh, Multidimensional Clustering Algorithms, *Lectures in Computational Statistics*, Vol. 4, Physica-Verlag, Würzburg, Germany, 1985.