

Semantic Crawling: an Approach based on Named Entity Recognition

Giulia Di Pietro, Carlo Aliprandi, Antonio E. De Luca, Matteo Raffaelli, Tiziana Soru

Synthema srl
Pisa, Italy

{giulia.dipietro, carlo.aliprandi, ercole.deluca, matteo.raffaelli, tiziana.soru}@synthema.it

Abstract—Law Enforcement Agencies (LEAs) are increasingly more reliant on information and communication technologies and affected by a society shaped by the Internet. The richness and quantity of information available from open sources, if properly gathered and processed, can provide valuable intelligence and help in drawing inferences from existing closed source intelligence. Today the intelligence cycle is characterized by manual collection and integration of data. Named Entity Recognition (NER) plays a fundamental role in Open Source Intelligence (OSINT) solutions when fighting crime. This paper describes the implementation of a NER-based focused web crawler under the EU FP7 Security Research Project CAPER (Collaborative information, Acquisition, Processing, Exploitation and Reporting for the prevention of organized crime). The crawler allows 1. to look for documents starting from a URL until a parametric depth of levels – also specifying a keyword that has to be contained in the page and in the related links – and 2. to look for a parametric number of documents starting from a keyword (entrusting the keyword search to one of the principal search engines, thus behaving as a meta-search engine). In addition, the crawler is able to retrieve only those documents that contain the information semantically relevant to the query (in other words: the required keyword with the required sense). This is achieved through the use of NER technologies. In this paper we present the CAPER NER-based Semantic Crawler, which has been proven to be a suitable tool for focused crawling, allowing LEAs to drastically reduce data collection and integration efforts.

Keywords—Open Source Intelligence (OSINT); Semantic Crawling; Named Entity Recognition (NER); Natural Language Processing (NLP).

I. INTRODUCTION

The revolution in information technology is making open sources more accessible, ubiquitous, and valuable. Law Enforcement Agencies (LEAs) have seen open sources grow increasingly in recent years and most valuable intelligence information is often hidden in files which are neither structured nor classified. The process of accessing all these raw data, heterogeneous in terms of source, format and language, and transforming them into information is therefore strongly linked to multi-modal and multi-lingual data analysis.

CAPER is an Open Source Intelligence (OSINT) platform that supports collaborative multilingual analysis of unstructured text and audiovisual contents (video, audio, speech and images) [1]. CAPER is not focused on the development of new technology, but on the fusion and real validation of existing state-of-the-art technologies to solve current bottlenecks faced by LEAs.

In this paper we present the crawler used in the CAPER platform. A crawler is a software that goes through the web, analyzing the URLs found in the web pages and making some operations on them: it can be used, in fact, to validate web pages, to create web indexes or to download web contents.

The crawler used in the CAPER project seeks to download web content to perform, in later steps, different kinds of analysis. In the next sections we will present the four different types of crawling available in the CAPER platform and then focus on an innovative crawling feature, which we will be referring to as Semantic Crawler.

We use the name of Semantic Crawler to make a clear difference between this crawler and the others which can be found in literature. In fact, it is worth here to make a distinction between *Focused Crawlers*, *Semantic-Web Crawlers* and the *Semantic Crawler* we propose in this paper.

Here is a brief description of these three kinds of crawler:

- **Focused Crawler:** this kind of crawler [2] follows a statistical approach. It is a semantically driven crawler too, but its focus is on the topic. It aims, in fact, to search and retrieve only the subset of the world-wide web that pertains to a specific topic of relevance. The gathering activity of a Focused Crawler is based on the evaluation of how much a page is relevant for a specific topic, computing some kind of similarity score between a model (representing the topic) and a singular page.
- **Semantic-Web Crawler:** this kind of crawler [3] is thought to crawl in a totally different way than traditional crawlers. Whereas a traditional crawler operates on HTML documents, linked using HTML anchors, a Semantic-Web Crawler operates on RDF

(Resource Description Framework) or OWL (Ontology Web Language) metadata in which links are implemented using the ‘RDF/OWL relationships’ [4].

- **Semantic Crawler:** the Semantic Crawler, on the other hand, works like a traditional crawler (analyzing and retrieving the URLs inside the HTML pages) but also includes a semantic layer of evaluation. It works like a meta-search engine, in which a keyword and the entity category that the keyword must have (*person, organization, location*, etc.) is specified (it can be considered as a NER-based meta-search engine). In this way, the crawler will be able to crawl only those web pages that are semantically relevant to the query.

Even though the Semantic Crawler could seem not to add any technological improvement to the crawling state-of-the-art, it may be very helpful inside the scope of the CAPER project. In fact, the Semantic Crawler is able to provide LEAs with a small set of documents which are very pertinent to those expected.

In the next sections we will first show what NER is, then we will see how the CAPER crawler works, and finally how NER is applied to the crawling process, giving life to the Semantic Crawler.

II. NAMED ENTITY RECOGNITION

NER is a subtask of information extraction, which aims to recognize and classify *entities* in texts. NER systems can recognize different entities based on the data sets they have been trained with. **Stanford NERC** (Named Entity Recognition and Classification), for instance, is provided with 3 different models, which give the possibility to use the same tool with different levels of complexity. They provide a 4 class model trained for the Conference on Computational Language Learning (CoNLL) [5] (whose entity classes are *Location, Person, Organization* and *Misc*, used for names of miscellaneous entities that do not belong to the previous three classes), a 7 class model trained for the Message Understanding Conference (MUC) (whose entity classes are *Time, Location, Organization, Person, Money, Percentage* and *Date*), and a 3 class model trained on both data sets for the intersection of those class sets (whose entity classes are only *Location, Person* and *Organization*).

In this work we use the NER tool developed inside the scope of the **OpenNER Project** [6]. It is based on Apache OpenNLP and has been trained on generic data. It is able to recognize and classify the following entity types:

- *Location*
- *Person*
- *Organization*
- *Misc*

This NER tool is available in 6 languages: *English, Italian, Spanish, French, Dutch* and *German*.

It is worth to note, as shown in Table 1, that the used NER tools achieve state-of-the-art performance.

TABLE I. PERFORMANCE OF THE NER TOOLS

	<i>Precision</i>	<i>Recall</i>	<i>F1</i>
Italian	81.15	62.70	70.74
English	89.39	85.19	87.20
German	84.01	58.56	69.01
Spanish	79.91	80.58	80.25
Dutch	79.85	75.41	77.57

III. THE CAPER CRAWLER

The crawler collects documents (we use the term document to refer to every textual source) from the Internet giving the possibility to *ask* for the kind of information that is needed. Our crawler [7] was already implementing three different crawling methods, which are:

- Crawling by Keyword
- Crawling by URL
- Crawling by URL focusing on Keyword

Even though there are some commonalities in the way they operate, each one of these methods works differently.

A. Crawling by Keyword

The **Crawling by keyword** method aims to get only those web pages containing a specified *keyword*. This method is the simplest one. It requires only 3 parameters: a *keyword*, the *number of pages* to be retrieved, and the *language* in which the page content is required. To get the pages to respond to all those parameters, the crawler strongly relies on the results gotten from a customizable search engine (in CAPER we use Google). The parameter *language* is required in order to make the search in the different Google domains. This means that, if the set language is *Italian*, the crawling will start from <http://www.google.it>, or if the language is *French*, it will start from <http://www.google.fr>, etc. The crawler allows the search in 14 languages, which are *English, Italian, Basque, Portuguese, German, French, Romanian, Japanese, Chinese, Spanish, Arabic, Hebrew, Catalan* and *Russian*. The *number of pages* parameter will define how many pages the crawler will have to get. Of course, the relevance of a page will be given by the Google page rank, which means that the pages crawled first are those pages which have been retrieved first by the search engine.

B. Crawling by URL

The **Crawling by URL** method aims to crawl all the documents it finds starting from a specified URL. It requires 3 parameters as well, which are the *urls*, *exclusion urls* and *depth*. The *urls* parameter, of course, is the most important parameter for this method, because it defines the URL which the crawler will start its web exploration from. The *depth* parameter is also very important for this method, because through it, it is possible to set how deep the crawler has to go analyzing also the URLs inside the page, as if it was a graph to be traversed. The *exclusion urls* is an optional parameter that can be helpful in defining which URLs have not to be crawled. This means that, if the crawler (during the web traversal) will end up in the URL set in the *exclusion urls* parameter, it will have to not crawl that page and to not proceed further with the traversal.

C. Crawling by URL focusing on Keyword

The **Crawling by URL focusing on Keyword** method can be seen as an enhancement of the Crawling by URL method. In fact, its goal is to keep crawling starting from a given URL and exploring all the links inside the web pages, but at the same time it also aims to crawl only those pages in which a given keyword appears. The required parameters are 4: the *keywords*, the *exclusion keywords*, the *url* and the *depth*. The parameters *url* and *depth* work exactly in the same way as in the Crawling by URL method. The *keywords* and *keyword that must not be in the page* are the parameters which will narrow down the results. In fact, every time a web page is explored by the crawler, a text check will be operated. If the *keyword* is found in the text – and no *exclusion keyword* is found –, the document will be considered satisfactory with regard to the set parameters, and the document will be crawled and stored.

D. Conclusion

The above described crawling methods also operate a *normalization* of the contents found on the web. Text files (which can be HTML or PHP files, but also PDF files) are in fact cleaned up, in order to extract only the valuable text. The CAPER crawler has been proven to be a suitable tool for helping LEAs reduce data collection, evaluation and integration efforts by 80% [7]. In addition, we have implemented a new crawling method, which we call **Semantic Crawler**, able to crawl only those documents that are semantically relevant to the information required for focused investigation.

IV. THE SEMANTIC CRAWLER

The **Semantic Crawler** aims to crawl only those documents that contain the information genuinely relevant for a specific investigation. Although the technology behind this system is not particularly complex, the Semantic Crawler is able to achieve the goal of crawling only documents that contain the sought information, allowing LEAs to spare time in analyzing only a small – but very pertinent – set of documents. The Semantic Crawler operates

Word Sense Disambiguation (WSD), in those cases in which a word can have more than one meaning. As an example, if LEAs are interested in having information about the hotel *Hilton in Paris*, instead of information about the model *Paris Hilton*, the crawling methods not performing semantic analysis will not allow such a focused crawling. This is not possible because, as mentioned previously, the **Crawling by keyword** method operates like a meta-search engine, basing the crawled documents on Google's page rank algorithm, and Google does not operate any kind of sense disambiguation. If we search this information on Google it is in fact impossible to get documents about only one entity (*Hilton in Paris*) or the other one (*Paris Hilton*). We operate this sort of disambiguation, choosing to crawl only those documents in which the words *Hilton* and *Paris* clearly indicate the *entity* we are searching for. In this case, *Hilton* is an **organization** and *Paris* is a **location**.

This method works in a very similar way to the Crawling by keywords. It takes as input four parameters, which are *keywords*, *senses*, *pages* and *language*.

The parameter *keywords* is, of course, a mandatory parameter. It can include one or more keywords which the search will be focused on.

This parameter is strictly related to the *senses* parameter, which is also mandatory in this kind of crawling. For each keyword it is possible to specify the entity the word has to be classified with.

The parameter *pages* specifies how many pages have to be checked. The crawler will be allowed to crawl at most the specified number of pages, verifying whether every single page contains the *keyword* with its relative *sense*.

The parameter *language* is required. Even though the crawler can operate in 14 languages, the OpeNER NER tools are only implemented for 6 languages, and among them Dutch is not implemented in the crawler. This means that the Semantic Crawler can so far be operated in 5 languages, which are *English*, *Italian*, *French*, *Spanish* and *German*.

Once all these parameters have been set, the crawling can be started. As already explained above, all the pages the crawler will get are based on a search engine search. In fact, the first thing this method takes into consideration is the set *language*, as it will perform the search in different Google domains depending on the set language. As an example, if the language set is English, the crawler will start the search from <http://www.google.co.uk>, whereas if the language set is Italian, the crawler will start the search from www.google.it, etc. Of course, the parameter which is then required to be set is the *keywords*. This parameter, which is a string, is combined with the search engine URL. As an example, if the user chooses to crawl documents about *Obama* in *Italian*, the URL retrieved by the crawler will be the following: <http://www.google.it/custom?&client=pub&hl=it&q=Obama>

In the link above the search engine used to get the documents is <http://www.google.it> and, of course, the language specified in the *hl* parameter (which refers to the

language) is *it*. The last parameter is *q* (which refers to the query). The specified keyword (in our example *Obama*) is attached at the end of the query parameter.

In case the *keywords* parameter contains more than one keyword, the crawler entrusts the search engine to manage the multiple keywords. In fact, this parameter has to be expressed using a specific syntax, which is actually the one used by the search engine. Even the *logical operators* which can be used to better express the information the user is willing to get must follow the search engine syntax.

At this point, the crawler will check the presence of *keyword* and *sense* in as many web pages as the number set in the *pages* parameter. Differently from what happens with the **Crawling by keyword** method, in which the crawled pages are the first pages returned by the search engine, in this case the crawler operates a sort of filtering, based on the presence of **Named Entities** in the text.

Every time a document is crawled and normalized, the crawler also extracts *Named Entities* from the text, using the OpenNER modules and operating the following NLP tasks:

- Language Identification (LI)
- Tokenization
- Part-of-Speech (POS) Tagging
- Named Entity Recognition (NER)

The generated output is a KAF (Knowledge Annotation Framework) [8] [9] file, which has an XML-like structure. In KAF files there are different annotation layers, each one of which encoding a different type of linguistics information. Once we have the KAF file with the *Named Entities* layer, we use a KAF parser to extract only that layer and the *Categories* the extracted Entities have been classified with. At this point, the crawler will check whether the keywords have been classified with the expected entities or not. In the first case the document will be kept stored in the repositories, otherwise it will be deleted, since it does not contain any information relevant to the query.

V. USE CASE

The Semantic Crawler has been extensively tested in Italian giving the expected results. Here follows an example of use case.

Let's suppose that Italian LEAs are interested in collecting information about a person named *Giuseppe Milano*. They set the Semantic Crawler as follows:

- **Keyword:** *Giuseppe Milano*
- **Sense:** *Person*
- **Pages:** 100
- **Language:** *Italian*

In such a case the Semantic Crawler is very helpful, given that *Milano* is both a family name and a city name.

Also, a hospital in Milan is named after San Giuseppe, which makes the search even more problematic. Using the simple Crawler by Keyword, LEAs would get all the documents in which the word *Milano* and the word *Giuseppe* (separately or together) occur, but it would be impossible to have any certainty that *Milano* is a surname and not the city, and that *Giuseppe* is a name and not the hospital.

Fig. 1 shows the different kinds of documents crawled by the **Crawling by Keywords** method.

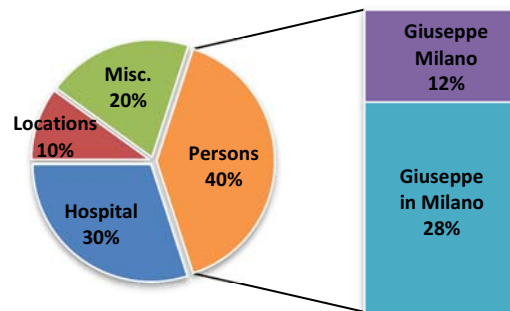


Fig. 1. Documents crawled using the Crawling by Keyword method

We asked the system to crawl 100 documents responding to the query *Giuseppe Milano*. As we can see, 10% of the crawled documents are about locations in Milan, such as streets and squares named after some Giuseppe; 30% of the documents are about the San Giuseppe Hospital in Milan; 20% of the documents talk about organizations (in Milan of course), schools or theatres, named after people called Giuseppe; 40% of the documents contain information about persons called Giuseppe, but only 12% of them have Milano as surname, whereas 28% of them are related to the city of Milan.

We have launched the same query using the **Semantic Crawler**. The system checked the 100 pages requested and downloaded only 10 documents, those in which *Giuseppe Milano* has been classified as a person. This happened for instance in the case of the soccer player *Giuseppe Milano* and in other few cases in which *Giuseppe Milano* is a doctor, a professor etc. The crawler did *not* download documents that deal with the *San Giuseppe Hospital in Milano*, or with *Via Giuseppe Meda in Milano*. The crawler was also able to avoid those documents in which a person named *Giuseppe* who lives in *Milano* is mentioned (or a person whose surname is *Milano*, but with a different forename).

That means that LEAs, that are interested in getting information about *Giuseppe Milano*, can spare a huge amount of time using the Semantic Crawler, having the possibility to read and analyze only the portion of documents they are actually interested in.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented one of the core elements of the CAPER system: the crawler. We have seen how it is possible to crawl documents from the web using three different crawling methods and we have presented the new Semantic Crawler. The strength of this new crawling method resides in the fact that it uses NER to solve the semantic ambiguity of a keyword and it is able to crawl only those documents in which the required keyword is contained with the required sense.

So far, we have extensively tested the Semantic Crawler only in Italian. Future work will involve the testing of the Semantic Crawler also in the other project languages. It is worth to note that the good results achieved in Italian are strictly related to the NER tool performance, therefore we expect to achieve similar results in all other project languages as well.

ACKNOWLEDGMENT

This work is partially funded by the European Commission (CAPER Project, 261712 FP7-SECURITY SEC-2010.1.2-1).

REFERENCES

- [1] C. Aliprandi, J. Arraiza Irujo, M. Cuadros, S. Maier, F. Melero and M. Raffaelli, "CAPER: Collaborative information, Acquisition, Processing, Exploitation and Reporting for the prevention of organized crime", Proceedings of the 16th International Conference on Human-Computer Interaction (HCI 2014), Crete, 2014 (forthcoming).
- [2] M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles and M. Gori, "Focused Crawling Using Context Graphs", Proceedings of the 26th VLDB Conference, Cairo, 2000.
- [3] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, P. Kolari, "Finding and Ranking Knowledge on the Semantic Web", Proceedings of the 4th International Semantic Web Conference (ISWC 2005), Galway, 2005.
- [4] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly et al., "Tabulator: Exploring and Analyzing linked data on the Semantic Web", Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI 2006), Athens, 2006.
- [5] Conference on Computational Natural Language Learning (CoNLL): <http://www.conll.org/>
- [6] R. Agerri, M. Cuadros, S. Gaines and G. Rigau: "OpenNER: Open Polarity Enhanced Named Entity Recognition", Sociedad Española para el Procesamiento del Lenguaje Natural, Volume 51, Madrid, 2013.
- [7] C. Aliprandi, G. Di Pietro, E. De Luca, M. Raffaelli, D. Gazzè and A. Rodríguez Rubio, "Data Acquisition", in P. Casanovas, J. Arraiza, F. Melero, L. Prenafeta (Eds.), "Crawling Open Source Intelligence. Collaborative Information, Acquisition, Processing, Exploitation and Reporting for the Prevention of Organised Crime", Chapter 4, Law, Governance and Technology Series, Berlin, Springer Verlag, in press.
- [8] M. Tesconi, F. Ronzano, S. Minutoli, C. Aliprandi and A. Marchetti, "KAFnotator: a multilingual semantic text annotation tool", Proceedings of the 5th Joint ISO-ACL/SIGSEM Workshop on Interoperable Semantic Annotation, in conjunction with the Second International Conference on Global Interoperability for Language Resources (ICGL 2010), Hong Kong, 2010.
- [9] W. Bosma, P. Vossen, A. Soroa, G. Rigau, M. Tesconi, A. Marchetti, M. Monachini and C. Aliprandi, "KAF: a generic semantic annotation format", Proceedings of the 5th International Conference on Generative Approaches to the Lexicon 2009, Pisa, 2009.