iWIN: a Summarizer System Based on a Semantic Analysis of Web Documents

Antonio d’Acierno  
ISA - CNR  
Via Roma, 64, 83100  
Avellino, Italy  
dacierno.a@isa.cnr.it

Vincenzo Moscato, Fabio Persia, Antonio Picariello  
DIS, University of Naples  
via Claudio 21, 80125  
Naples, Italy  
{vmoscato,fabio.persia,picus}@unina.it

Antonio Penta  
ECS - University of Southampton  
Hampshire SO17 1BJ  
Southampton, UK  
ap7@ecs.soton.ac.uk

Abstract—Seeking bits of useful information from a large amount of data on the Web still remains a difficult and time consuming task for a wide range of people such as students, reporters, and many other types of professionals. This problem requires to investigate new ways to handle and process information, that has to be delivered in a rather small space, retrieved in a short time, and represented as accurately as possible. This is surely one of the most important reasons for searching suitable and efficient summarization techniques capable of “distilling” the most important information from a variety of logically related sources, as the one returned from classic search engines, in order to produce a short, concise and grammatically meaningful version of information spread out in pages and pages of texts. In this paper we present a summarizer system, named iWIN (information on the Web In a Nutshell), that is able to perform an automatic summarization of multiple documents through: a semantic analysis of the text, a ranking method used to evaluate the relevance of the information for the specific user, a clustering method based on the document representation in terms of set of triplets (subject, verb, object) and a sentences’ selection/ordering process to make the final summary as much readable as possible. Some preliminary results about system performances obtained using the ROUGE evaluation software are presented and discussed.

Keywords—Summarization; NLP; Semantic Analysis.

I. INTRODUCTION

The exponential growth of the Web has made the search and the track of information apparently easier and faster, but the huge information overload requires algorithms and tools able to allow a fast and easy access to the specific desired information. desired information.

Indeed, it seems strategic to research (novel) ways to handle and process different kinds of information, that has to be delivered in a rather small space, retrieved in a short time, and represented as accurately as possible. In this challenging framework, particularly relevant are the so called summarization techniques, where the summarization represents the process of filtering, highlighting and organizing information from a variety of logically related sources in order to produce a short, concise and grammatically meaningful version of information spread out in pages and pages of texts.

In this paper, we propose a novel approach for text summarization based on semantic analysis of documents.

At a glance, we propose to capture the semantic nature of a given document, expressed in natural language, by a set of triplets (subjects, verbs, objects related to each sentence) and to cluster these ones thus aggregating similar information. In our vision, the triplets’ statements can be considered as the basic unit in the process of summarization and the more similar the triplets are, the more the information is useless repeated; thus, a summary may be thus constructed using a sequence of sentences related to the computed clusters.

To better explain our idea, let us consider a reporter that has been just informed about the crash of a piper in Milan and that quickly would like to gather more details about the event and the following textual information are publicly available on the Web. By means of a standard engine, our reporter finds the following documents: “A Rockwell Commander 112 airplane crashed into the upper floors of the Pirelli Tower in Milan, Italy. Police and ambulances are at the scene. The president, Marcello Pera, just moments ago was informed about the incident in Milan, he said at his afternoon press briefing”.

“It was the second time since the Sept 11 terror attacks on New York and Washington that a plane has struck a high-rise building. Many people were on the streets as they left work for the evening at the time of the crash. Ambulances streamed into the area and pedestrians peered upward at the sky. The clock fell to the floor. In Rome, a spokesman for the senate president, Marcello Pera, said the interior minister had informed him that the crash didn’t appear to be a terror attack”.

To automatically summarize such documents, we first propose to reduce each sentence to a list of triplets formed by (subject, verb, object), the verb being reported in the infinitive form. We then propose to clusterize the triplets, using a semantic similarity function between each pair of triplets. Our vision is that sentences in the same cluster will contain similar information and that we can obtain a useful summary just considering the sequence of each centroid, eventually re-loading the associated original sentence.

With reference to the previous air crash example, Table I reports the extracted triplets grouped in 3 clusters; underlined triplets represent the three centroids that allow to obtain the following summary:
A Rockwell Commander 112 airplane crashed into the upper floors of the Pirelli Tower in Milan, Italy. Police and ambulances are at the scene. In Rome, a spokesman for the senate president, Marcello Pera, said the interior minister had informed him that the crash didn’t appear to be a terror attack.

<table>
<thead>
<tr>
<th>Cluster 1</th>
<th>(ambulance, be, scene)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(police, be, scene)</td>
</tr>
<tr>
<td></td>
<td>(ambulance, stream, area)</td>
</tr>
<tr>
<td></td>
<td>(people, be, street)</td>
</tr>
<tr>
<td></td>
<td>(pedestrian, peer, sky)</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>(minister, inform, crash)</td>
</tr>
<tr>
<td></td>
<td>(president, inform, incident)</td>
</tr>
<tr>
<td></td>
<td>(spokesman, say, minister)</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>(airplane, crash, tower)</td>
</tr>
<tr>
<td></td>
<td>(plane, strike, building)</td>
</tr>
<tr>
<td></td>
<td>(clock, fall, floor)</td>
</tr>
<tr>
<td></td>
<td>(terror, attack, Washington)</td>
</tr>
<tr>
<td></td>
<td>(terror, attack, New York)</td>
</tr>
</tbody>
</table>

Summing up, in this paper we present a novel summarization system called iWIN (information on the Web In a Nutshell), that provides to: (i) search relevant documents for a given retrieval task, (ii) collect pieces of information, (iii) plausibly rank the relevance related to the pieces of information, (iv) compose the pieces of information in a summary, (v) deliver the summary.

In other words, our system performs an automatic summarization of multiple documents through: a semantic analysis of the text, a ranking method used to evaluate the relevance of the information for the specific user, a clustering method based on the triplets’ representation and a sentences’ selection/ordering process to make the final summary as much readable as possible.

The paper is organized as follows. Section 2 consists of a schematic literature overview on summarizer systems. In Section 3 we discuss a general model useful to solve the problem of automatic summarization. Section 4 presents the summarization framework developed to test the model. Section 5 is dedicated to the experiments we performed to test the system performances. Finally, section 6 summarizes the results of this work and proposes some future developments.

II. RELATED WORK

Summarization, i.e. the process of producing a shortened version of a document or a set of documents, has been studied for a long time in the scientific community.

In particular, automatic document summarization has been actively researched since the original work by Luhn [1] in 1958 and, nowadays, the automatic summarization can be considered both a problem of computational linguistics and of information retrieval [2].

The use of computational linguistics techniques is due to the need to make computers able to process texts in a human readable form. Moreover, the automatic summarization is a valid approach to information retrieval because it allows to retrieve information from text documents (like web pages) in a short time and in a reliable way.

In the literature [2], automatic summarization systems are classified by a number of different criteria.

Usually in terms of goals, we have the following classification: (i) generic - general purposes summaries, where the goal is to represent all the information available in the source documents; (ii) query-based - summaries shaped by a user-query; (iii) topic or user focused - summaries that gives more importance to information about a given topic of interest and/or considering the profile of a specific user.

Another main distinction is between summarization of a single or multiple documents. In the latter, we have multiple inputs into the systems, this brings more challenges, because it has to take into accounts different aspects such as the diversity and the similarity of different sources and the temporal order of the extracted information. The main examples of multi-summaries systems are the ones that present HTML documents in a condensed form. Furthermore, we have also a distinction between abstraction-based or extraction-based summary: extraction-based summary involves the selection of text fragments from the source document while the abstraction-based summary involves the generation of new texts.

Moreover, we may differentiate automatic summarizer systems by the type of summary they generate. In the literature, the most diffused classification is the following: (i) indicative - summaries provide enough information to alert a user about sources that can be read in more depth; (ii) informative - the summary substitutes the related sources providing the relevant pieces of information in a concise structure; (iii) critical - such summaries incorporate opinion statements on content.

Despite the described differences, the main issues of a summarization system lie in the following two main tasks: how to determine the salient text fragments in the sources and how to reduce or reproduce them by preserving the most useful information.

A key factor of both tasks is to understand how the information can be considered relevant; it is not possible to state in an absolute way the relevance of the information, but we can understand what information a person needs or what information plays a main role in an article or a document. Therefore summaries, likewise the systems that generate summaries, differs by what they consider relevant and how they measure the relevance of the information. Here, we propose a generic, extraction-based, indicative and multi-document summarization technique.
Table II
OVERVIEW OF SOME SUMMARIZER SYSTEMS

<table>
<thead>
<tr>
<th>System</th>
<th>Purpose</th>
<th>Input</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The Table II shows an overview of the latest recommender systems that have some similarity to our approach. In particular: (i) we use a NLP processing pipeline to extract relevant information from texts similarly to [3], [9]; (ii) we introduce a semantic analysis of extracted text following the suggestions in [4], [6], [8]; (iii) according to [7], we use a clustering method to group similar information; (iv) and finally we adopt a sequence selection algorithm based on the maximum coverage problem as in [5], [10].

Eventually, a similar methodology to ours is proposed in [14], where the authors propose an extractive summarization of ontology based on a notion of RDF sentence as the basic unit of summarization. An RDF Sentence Graph is proposed to characterize the links between RDF sentences derived from a given ontology and the salience of each RDF sentence is assessed in terms of its centrality in the graph. The summarization is obtained by extracting a set of salient RDF sentences according to a re-ranking strategy.

III. A SUMMARIZATION MODEL FOR MULTIPLE DOCUMENT SOURCES

The summarization problem may be stated as follows: given a set of source documents, let produce an accurate and all-sided summary that is able to reflect the main concepts expressed by the original documents, matching some length restrictions and without introducing additional and not required information.

A summarization process based on an extractive approach is usually performed in two different phases.

The first one consists of some operations to extract those parts of text that convey the most important information contents; the second phase arranges the extracted text, trying to improve summary coherence and readability.

We think that the detection of the “semantic structure” of a text is at the basis for the extraction of the main concepts that can be added to a summary, whose meaning is a reliable and accurate synthesis of the original sources, thus eliminating redundancy.

The starting point of our theory is the concept of Summarizable Sentence \( \sigma \) defined over a document \( D_i \), as in the following:

\[
\sigma = \langle s, (t_1, t_2, \ldots, t_m) \rangle
\]

\( s \) being a sentence belonging to \( D_i \) and \( (t_1, t_2, \ldots, t_m) \) being a set of atomic or structured information that expresses the semantic content related to \( s \).

In particular, we consider \( t_i = \langle A_i, V_i \rangle \), \( A_i \) being any relevant atomic or structured attribute on which a generic classifier is trained and \( V_i \) a string or a set of strings in the sentence that is classified as a value for \( A_i \). Let us call \( t_i \) Semantic Unit.

A Summary (\( S \)) of a Summarizable Document Set \( D = \ldots \)
\{D_i\}; i = 1...N is a set of sentences coming from the original documents such that the following conditions are verified:

\[ \Lambda(S) \ll \sum_{i=1}^{N} \Lambda(D_i) \]  
\[ \mathcal{J}(S) \subset \bigcup_{i=1}^{N} \mathcal{J}(D_i) \]

\( \Lambda() \) being a function that returns the number of summarizable sentences and \( \mathcal{J}() \) being a function that returns the Information Set, i.e., the set of all semantic units related to a set of summarizable sentences.

We named the properties, expressed by eq. 2 and eq. 3, Length Restriction Condition and Accuracy Restriction Condition respectively, \( \Lambda() \) as Length Function and \( \mathcal{J}() \) as Information Set Function. Moreover, we note that in the previous definition, we implicitly assume that \( \Lambda(\emptyset) = 0 \) and \( \mathcal{J}(\emptyset) = \emptyset \).

In other terms a summary is the subset of summarizable sentences that has to be shorter than the union of the original documents and does not add new inormative concepts from the original set of documents.

There exists a lot of very different ways to summarize document set, and, to the best of our knowledge, there is no way to determine objectively if a summary is better than another: the quality of a summary is strictly connected with its information content and there is no method to evaluate the absolute relevance of information either. Therefore, what we did is to try to define some requirements. For example, we are looking for a summary that contains the information content of the source documents as faithfully as possible but, at the same time, we would eliminate text redundancy. A summarization technique generally aims at generating a summary whose information set has the maximum number of relevant information in terms of semantic units with respect to the original documents, also noted in literature as the Maximum Coverage Problem [16].

We can solve the problem of detecting the most representative content of a summary scoring each related information set. In particular, each semantic unit differently contributes to convey the semantic content of the source documents. Therefore, we want to consider a quantitative representation of such difference introducing the concept of Semantic Unit Score Function \( \Omega_t : t \rightarrow \omega_t \), \( t \) being a semantic unit and \( \omega_t \in [0, 1] \). This function is useful to associate a weight or importance to each semantic unit. In a similar way, it is possible to define an Information Set Score Function \( \Omega_S : \mathcal{J}() \rightarrow \omega_S : \omega_S = f_S(\omega_t), \mathcal{J}() \) being a given information set, \( \omega_S \in [0, 1] \) and \( f \) a function that takes into account both the aggregation values of single semantic unit scores (\( \text{max}, \text{min}, \text{avg} \)) and the number of such units. This function is useful to associate a weight or importance to an information set.

Assuming the existence of a function \( \gamma(t_i, t_j) \in [0, 1] \) able to compute the semantic similarity between two semantic units, we can introduce a function for measuring a similarity distance between a semantic unit and an information set, in order to understand if a summary effectively contains the most representative content of a set of documents:

\[ d(t, \mathcal{J}(\cdot)) = 1 - \max_{t_i \in \mathcal{J}(\cdot)}(\gamma(t, t_i)) \]

\( t \) being a semantic unit and \( \mathcal{J}(\cdot) \) an information set.

In other words, in our model a generic summary \( S \) is a set of sentences coming from a set of documents \( \mathcal{D} \) that: (i) has to reflect the semantic content of \( \mathcal{D} \), (ii) has to satisfy Length and Accuracy Restriction conditions. In the following, a Summarization Algorithm for generating a summary matching the above described conditions is presented.

Let \( \mathcal{D} \) be a set of summarizable documents, a summarization algorithm is a sequence of two functions \( \phi \) and \( \chi \). The first one \( \phi : (\mathcal{J}(\mathcal{D}), K) \rightarrow (\mathcal{T}_1, \tau_1), \ldots , (\mathcal{T}_K, \tau_K) \), named Semantic Clustering, partitions \( \mathcal{J}(\mathcal{D}) \) in \( K \) sets \( \mathcal{T}_1, \ldots , \mathcal{T}_K \) of information sets having similar semantics and returns, for each set, the related score and has to satisfy the following constraints:

1) \( K \ll \sum_{i=1}^{N} \Lambda(D_i) \),
2) \( \tau_k = \Omega_S(\mathcal{T}_k) \),
3) \( \mathcal{T}_{k_1} \cap \mathcal{T}_{k_2} = \emptyset, \forall k_1 \neq k_2 \).

The second one \( \chi : (\mathcal{T}_1, \ldots , \mathcal{T}_K) \rightarrow S \), named Sentence Selection, aims at selecting a set of the sentences from original documents that contain the semantics of more important clustered information sets such that:

1) \( \Lambda(S) \leq K \),
2) \( \forall \mathcal{T}_k : \Omega_S(\mathcal{T}_k) \geq \tau, \exists t \in S, d(t, \mathcal{T}_k) = 0 \).

\( \tau \) being a given threshold used to select the more important clusters. It is possible to note that, with abuse of notation, we use the symbol \( \in_S \) to indicate that a semantic unit comes from a sentence belonging to a set of summarizable sentences.

Starting from the described model the more important issues are related to: (i) how to represent semantic units of a document, (ii) how to calculate a score for each unit, (iii) how to evaluate the similarity between two semantic units, and eventually, (iv) how to define a possible summarization algorithm.

To give an answer to the first problem, we decided to adopt the Resource Description Framework (RDF), used in the Semantic Web community to represent data in machine readable format, to link different data, and to express the related semantics.

In other terms, we assume that the semantic content of a document is modeled by a set of structured information \( \mathcal{I}=(t_1, \ldots , t_n) \), where \( t_i = \{(\text{subject, value}), (\text{verb, value}), (\text{object, value})\} \) is an explicit way to serialize an RDF triple.
As similarity function among the corresponding “disambi-
guited” elements (synsets) of the triples, we decided to
use a function based on the semantic coincidence and in

The similarity partial values obtained for the couples
of subjects, verbs and objects of two sentences are then
combined by means of a classical weighted arithmetic mean.

As semantic unit score function regards, such a score is
obtained by combining: (i) a value of reputation that depends
on the value of reliability of the related source to which
the semantic unit belongs; (ii) the frequency of the semantic
unit, in terms of occurrences of the related subject, predicate
and object across the document set; (iii) the position of the
sentence, to which the semantic unit is associated in the
original document; (iv) the semantic importance with respect
to a specified and possible query or a topic.

Finally, as summarization algorithm, we could select any
unsupervised clustering algorithm able to partition the space
do documents into several clusters, where the number of
clusters is not defined a-priori, on the base of the discussed
semantic similarity among semantic units.

Once obtained a partition of the space in terms of clusters
of semantic units, we have to select the sentences that will
constitute the final summary trying to: (i) maximize the
clusters’ coverage - the more representative sentences of
each cluster should be considered starting from the more
important clusters, i.e. those having the highest average
information score; (ii) minimize the clusters’ redundancy -
during the selection of sentences (that can be related to more
than one cluster) we avoid to select a cluster more than one
time.

In our work, we have experimented the OPTIC algorithm
[12]. In particular, we repeated the execution of the OPTIC
algorithm using several settings in order to obtain a good
number of candidate summaries of different lengths.

Then, we use a sequential sentence selection algorithm
[16] based on the maximum coverage and the minimum
redundancy to choose the final sentences.

Once the optimal summary has been determined, the
sentences are ordered to maximize the partial ordering of
sentences within the single documents.

In other terms if there are two or more sentences in the
summary that come from the same document, they have to
maintain the same original order.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION
DETAILS

Our summarization process is described in the following.

1) Textual sentences are extracted from web pages by
parsing the related HTML code. In particular: (a)
useful text is detected by analyzing HTML tags; (b)
the sentences are extracted and the Named Entities
related to each sentence are recognized; (d) anaphora
crossing references and named entities co-references
are opportunistically solved and a semantic normalization
is performed [15].

2) Subject, predicate and object of each extracted sen-
tence are detected analyzing the related Parse Tree.

3) The discovered subject, predicate and object are ex-
tracted from each sentence and represented in a
subject-predicate-object space by a RDF format.

4) A matrix containing the semantic distance for each
couple of summarizable sentences is computed.

5) A clustering algorithm is applied as first step of
the summarization algorithm in the subject-predicate-
object space and the more important sentences are
selected;

6) The summary is selected in according to the semantic
coverage condition;

7) The sentences related to the representative of each
clusters of the optimal summary are then ordered and
organized in a summary.

The architectural pattern chosen for our system to support
the described process is based on the “pipe line and filter”
model. In particular, the system workflow proceeds accord-
ing to a sequential fashion characterized by data streams that
go through a set of information filters.

The component diagram of our system is reported in Fig.
1. The system is characterized by components that imple-
ments the functionalities related to the different steps of
the summarization process. Moreover, the system provides a
graphical user interface that allows the user to interact with
the system through graphical elements.

Our summarization system is quite similar to an Internet
search engine: a user, by means of several keywords, acti-
vates a search engine that stores into a local repository all
the web pages satisfying the search, by means of a WEB
Interface component. We have used the standard Google
API. These pages are then processed and the related textual
sentences of the document are extracted using the Jsoup
HTML Parser. In a second phase, several NLP algorithms
split the extracted text into sentences; a Named Entities
Recognition (NER) process followed by a part of speech
(POS) tagging is then applied by the Stanford NLP Libraries.

Successively, the related semantic content from each
original sentence in terms of RDF triples is extracted by
analyzing the sentence Parse Tree.

The tree generation is obtained using the Stanford Parser,
while the tree inspection is realized by the Stanford Tregex
Libraries.

A clustering of the RDF triples is then executed, and in
our implementation, we used the OPTIC clustering algo-

rithm and adopted as semantic distance the Wu & Palmer
distance. Eventually, a dedicated component has the task of
generating the summary related to the search keywords and
user preferences. In particular, for each clustering instance,
more important sentences on the base of generated clusters
(the most important sentences are those related to the cluster
having the highest information scores) are selected using the sentence selection algorithm, ordered and organized in the final summary.

The Triplets Extraction component gets as input the enhanced text generated by the NLP module and identifies the functional elements (subject, verb and complement) of each sentence thus extracting the related triplet. The module uses the Stanford Parser to generate, from a certain sentence, the dendrograms representing the grammatical structure of the text. In the particular implementation that we have adopted, the Stanford Parser generates the tree structure according to the Penn Treebank Project [13]. In the following we show an example of the most frequently used search pattern to identify subjects and complements related to each verb in the sentences.

1) A parse tree is generated for each sentence. For example, Figure 2 shows the parse tree related to the sentence “A small plat has hit a skyscraper in central Milan”.
2) Apposite heuristic search patterns (regex), based on
relationships among nodes and on their labels, are applied on the generated sub-trees to identify the semantic structure of a sentence. In Figure 2, the selected heuristics tries to identify the descendants of the subject node and the descendants of the verb node in order to discover subject, predicate and object.

V. PRELIMINARY EXPERIMENTAL RESULTS

In this section the experimental protocol and the obtained results are described and discussed.

Generally, evaluation of text summaries has been discussed for years and summarization technologies are evaluated by measuring the agreement of their extracted summaries with respect to human-generated summaries, which are called ground truths.

Many evaluation metrics have been proposed to measure the performance of a summarization approach and these metrics can be classified into three main categories: recall-based, sentence-rank-based and content-based.

In this work, we present an evaluation of our summarization model, based on ground truth summaries produced by human experts, using recall-based metrics. In particular, in order to compare the performance of our system w.r.t. other ones, we employed ROUGE (Recall-Oriented Understudy for Gisting Evaluation), an automatic summarization evaluation package.

More into details, in order to obtain a preliminary evaluation of our system, we decide to test the effectiveness of our approach on the data set of documents related to the DUC 2007 Workshop, that is one of the most recent challenges whose results are fully available.

In particular, for this kind of problem, 10 NIST experts have written summaries on about 45 topics and for each topic, chosen by a “topic selector”, there are 4 summaries that are manually generated by 10 different “human summarizers” identified by A-J letters. The table III shows the chosen topics for our experimentation and the related ground truth.

During the design and development phases, we note that the system performances are mainly affected by the similarity values among triples and by some clustering parameters. In particular, during the computing of semantic similarity between two sentences, we observed that it is critical the choice of: (i) the assigned weight for the subjects’ similarity ($\alpha_s$), (ii) the assigned weight for the verbs’ similarity ($\alpha_v$), and (iii) the assigned weight for the objects’ similarity ($\alpha_o$). Thus, on the base of a ROC analysis, we decided to use four main parameters settings.

For what the clustering concerns, in a similar way we have detected in the $\epsilon$ parameter [12] of the OPTIC algorithm the more important variable and choose for it three main settings. Thus, the possible system configurations, obtained by the combination of the similarity and clustering settings are listed into Table IV.

In order to choose the best configuration, we computed system performances in terms of average recall, average precision and F-measure with respect to the ROUGE-2 and ROUGE-SU4 metrics. Considering recall as the more important measure, in Table V the best average system scores and the related configurations are shown.

Finally, on the base of selected configuration, Tables VI and VII show the position of the implemented system in DUC 2007 global ranking.

We can observe that although the system has not been de-
signed for a query-based approach as in DUC 2007 requirements, the obtained performances are quite encouraging. Furthermore, it is possible to note how the results show that our system can be considered better than CLASSY04 ID=2, that is the generic automatic summarizer which reached the highest score in the multi-document summarization field of DUC 2004 workshop.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we described a novel approach for summarizing web documents based on semantic extraction and RDF description of documents. In particular, we proposed a system able to build summaries by using sequences of clusters samples in the RDF space.

We tested our system using some well known data corpora and we obtained promising results with respect to other approaches.

Further works will be devoted to improve our work into main directions: i) extend the methodology to multimedia data; ii) design more detailed experiments and iii) compare the results with other similar summarizer systems.

ACKNOWLEDGMENT

This research has been partially funded by the FLAGSHIP InterOmics project (PB.P05, supported by the Italian MIUR and CNR organizations).

REFERENCES


