Semantic Analysis Applications in Computational Bibliometrics

Sorin Avram *, Victor Velter, Ioan Dumitrache***

** University Politehnica of Bucharest, 060042*
Romania (e-mail: avram.sorin@gmail.com, ioan.dumitrache@acse.pub.ro)
*** Executive Agency for Higher Education, Research, Development and Innovation*
Funding, Bucharest, 010362, Romania (e-mail: victor.velter@uefiscdi.ro)

Abstract: Continuing a previous theoretical research in bibliometrics, this study aims to conclude a bibliometric endeavor, in the quest of finding an adapted impact measure for scientific papers. Its main objective is to define a technological solution capable to interpret both citations and papers' content, in an integrative approach. The solution employs natural language processors, similarity measures and graph computation algorithms, while integrating them in a software prototype. Describing the design and implementation phases, the research underlines specific solutions and optimizations for relevance computing in citation networks.

Keywords: bibliometrics, citation weighting, natural language processing, text similarity

1. INTRODUCTION

The starting point for the current study rests in a previous theoretical research „A new approach to bibliometrics based on semantic similarity of scientific papers" (Avram *et al.,* 2012) that presents the recent evolution in the field of scientific information assessment, proposing a bibliometric measure for the relevance of the scientific papers. Evaluating scientific publications has proved to be a challenging task, given the continuous evolution and specific citation patterns of each research field. We now have nearly 50 years of bibliometric research, but the majority is focused on journals, proceedings and more recently, books.

The objectives of the present research are the design and implementation of a technological solution, capable to support the computing of an article focused metric, processing and employing the entire informational universe of a paper: citations and scientific content. From the technological point of view, implementing the software prototype is a cross-disciplinary endeavor, using state-of-the-art technology for robust, broad-coverage natural language processing and specific methods from Data Mining – Information Retrieval.

The study is in-line with the latest trends in bibliometrics, incorporating different sources of relevant data and going beyond the citations analysis, a research field that is already inflated by countless indicators, rankings and niche applications.

Bibliometrics are still widely used as a generic term for the correlated fields of sciento/info/techno-metrics where publications are considered the elementary units of scientific information and the main source of indicators. The diversity of new patterns of communication on the electronic network blurs sometimes the frontiers between formal and informal circulation, between activities taking place inside and outside 'science' (Heimeriks *et al.,* 2002). In this context, bibliometrics is thus experiencing a revival, not primarily with respect to mathematical modeling and theoretical principles, but as an instrument of science management (Ball *et al.,* 2006).

This paper is structured in 6 sections, presenting the need for an article relevance measure, development phases, specific optimizations and observations over the prototype implementation and conclusions after testing the applications with a subset of scientific articles.

2. ARTICLE BASED INDICATORS

Bibliometric indicators seek to measure the quantity and impact of scientific publications and are based on a count of scientific papers and the citations they receive. Together with patent indicators, they are one of the most frequently used indicators of research and experimental development (R&D) output. From a qualitative perspective, most of the bibliometrics indicators are focused on journal ranking: Thomson's Impact Factor, Eigenfactor (EF), Article Influence Score (AIS), Cited Half-Life (CHL), and Elsevier's SCImago Journal Rank (SJR), Source Normalized Impact per Paper (SNIP). While this has been an important drawback in the use of bibliometrics, only a few attempts to implement an article based indicator were made, with little or no success. An important signal was given by the UK Research Excellence Framework (REF), when discussing the use of citation data in the research evaluation of published articles, giving the community a chance to innovate and propose an enhanced and adapted solution (Levitt *et al.,* 2011).

2.1 Why do we need an article focused bibliometric metric?

1) The popular „Article Influence Score" (AIS), computed by Thomson Reuters and published in bibliometric catalog Journal Citation Reports (JCR) doesn't cover all the publications available in the catalog (Braun *et al.,* 2010).

2) Even though the name „Article Influence Score" might indicate a metric for the article scientific influence, the index

cannot discriminate between the values published in the same journal.

3) The only available bibliometric measure for articles is the citation number, available in different bibliographic databases like Thomson Reuters's JCR, Elsevier's Scopus and Google Scholar with its latest product „Publish or Perish" (Velter, 2010).

4) After a close overview on the ISI rankings (for journals indexed in Thomson Reuters JCR) for both the Impact Factor and the Article Influence Score, one can observe very big differences in scores and rank position for the same journal. Table 1 presents a relevant sample from the parallel study of AIS an IF indexes: journal ranking positions are calculated within the corresponding research field, as they are present in the Web of Science – Categories catalog.

Table 1. Sample for ranking comparison between Article Influence Score and Impact Factor

AIS Rank	IF Rank	JOURNAL
1	9	Reviews of Modern Physics
2	10	New England Journal of Medicine
3	6	Nature Photonics
4	19	Chemical Reviews
5	3	Nature
6	4	Science
7	5	Nature Materials
8	61	Nature Physics
9	33	JAMA - Journal of the American Medical Association
10	12	Nature Nanotechnology
11	26	Lancet
12	2	A Cancer Journal for Clinicians
13	17	Annual Review of Plant Biology
14	21	Annual Review of Immunology
15	8	Annual Review of Psychology
7778	82	Materials Today
7779	161	Journal of Computer Mediated Communication
7780	168	Who - Technical Report Series
7781	185	Living Reviews in Relativity
7782	226	Progress in Electromagnetics Research – Pier
7783	250	Transport
7784	260	Abstract and Applied Analysis
7785	284	Journal of Web Semantics
7786	318	Trauma Violence & Abuse
7787	337	Academy of Management Learning & Education
7788	368	Strategic Organization
7789	482	Alternative Medicine Review
7790	507	Review of Research in Education

2.2 Article Relevance Factor

Published in a first integrated research, „A new approach to bibliometrics based on semantic similarity of scientific papers", the Article Relevance Factor defines a new measure for scientific relevance, calculated at article level (Avram, et al., 2012):

$$ARF(a) = (1-d)\frac{CC(a)}{\sum_{j=1}^{N} CC(a_j)} + d\sum_{i=1}^{k} ARF(a_i) * SSW(a_i, a) \quad (1)$$

The advantages of the metric, as they were already presented in (Avram *et al.,* 2012), provide a strong base for further implementation and adoption in large production systems:

- journal size independence;
- research field pattern and citation frequency independence;
- capable of employing the whole citation network;
- capable of interpreting the content of the papers and to generate the corresponding citation weight based on the scientific relevance;
- resistant to citation engineering practices (empty-citations, citation-clubs).

The computational effort for the ARF metric can be sequenced in 4 generic phases, based on the transformations and operations that the information is going through:

1) content semantic processing and conceptual structure extraction;
2) computing TF-IDF values for extracted concepts;
3) computing the citation relevance weight (CRW) and the semantic similarity weight(SSW);
4) ARF computing, using the iterative Pagerank algorithm, adapted to citation networks.

3. SCIENTIFIC DISCOURSE ANALYSIS – CONCEPTUAL STRUCTURE EXTRACTION

The first processing phase aims to extract the conceptual structure of each scientific article, to provide the input data for calculating concepts frequencies weights, in the TF-IDF computing phase.

Extracting the concepts from each text, as it's been proved in the case study of (Avram *et al.,* 2012), involved the integration of a natural language processing (NLP) tool. As a few tools for NLP, capable to support different types of text processing and different programming languages, are already available on the market, the decision in terms of prototype implementation came down to choosing the optimal one.

3.1 Stanford Core NLP and Apache Core NLP libraries

Two of the most appreciated and well-known tools in the field are the Stanford Core NLP and the Apache Open NLP; while the first is created by a group of researchers led by Prof. Chris Manning, from the famous Californian university (Stanford University), the second is an open-source initiative within the Apache Software Foundation (The Apache Software Foundation, 2010). In a more comprehensive evaluation, Ievgen Karlin (Karlin, 2012) describes the differences between the two, underlining the advantages and functionalities of Core NLP over the open-source alternative, as they are presented in table 1.

Table 2. Abilities of Open NLP and Core NLP

(Karlin, 2012)

Ability	Stanford Core NLP	Apache Open NLP
Sentence Detection	+	+
Token Detection	+	+
Lemmatization	+	-
Part-of-speech Tagging	+	+
Named Entity Recognition	+	+
Co-reference Resolution	+	-

In terms of dictionary cover, the lemmatizer offered by the Core NLP toolkit outputs 142,293 lemmas, also superior to the Open NLP (Ryzko *et al.,* 2011). Also, in terms of usability, Core NLP is available in different packages, for the most common programming languages: Java, Perl, Python and Ruby. The library offers an integrated platform of text analytical tools, being capable of recognizing words dependencies, to clear out word senses ambiguities and to recognize composed language structures. Another important function of the Core NLP is recognizing word dependencies and grammatical structures, so that using the dictionary it can output the concepts from the text, represented by nouns.

Having selected Stanford Core NLP as the tool for the semantic processing phase, the implementation followed the steps required for engine setup and running: using a dedicated *java properties* structure, Core NLP is loading the four *annotators*, which are the functional classes for text processing (The Stanford Natural Language Processing Group, Core NLP Tools).

1) *Tokenize* - This processing class uses a PTB (Penn Treebank 3) algorithm, in an extended version, to handle noisy text and web pages.
2) *Sentence Split* – is a processing class that takes the tokenized text and marks the beginning and the end of each sentence.
3) *Part Of Speech Annotation* – is a tagger class, capable of parsing the text and tagging each word (token) with its morphological class, number etc.
4) *Lemma* – is a functional class that returns the lemma (the canonical or the citation form) of each word.

A decisive step in the extraction process is lemmatization. After identifying the concepts in each text, using them in their form might have caused diluted results. A concept appearing in derived forms (e.g. book, books) would have been registered with different frequencies for each form, while having the same semantic value. A lemmatization of each word is mandatory in order to have consistent results when computing the similarity degree.

The output of this phase is a matrix, having the following coordinates: each article from analyzed corpus D has a corresponding line ($||D|| = N$), while each column is corresponding to a concept (from the set of extracted concepts). Each cell of the matrix, *concept(i,j),* hosts the frequency value of concept i in article j.

3.2 Tropes – a multi-language NLP processor

As Core NLP and Open NLP offer text processing capabilities only for English documents, there was an obvious need for a broader, more customizable and largely applicable solution.

Examining the opportunities to integrate a semantic processing module, the authors evaluated the solution of using an external application: Tropes V8.3. Appeared in the first version in 1994, Tropes was created by Pierre Molette in partnership with the University of Paris VIII, achieving cognitive processing, lexical, semantic and results extraction in various graphs, reports or specific data structures (Molette, 2012). Being available for six international languages, Tropes had another important advantage, of being a standalone, optimized software application.

In a first phase, the application indexes the articles present in the citations network, whether they are citation sources or cited articles. Their selection was based on the initial results from text conversion and citations identification. The limitations of Tropes are only related to hardware configuration: memory size and storage capacity. Tropes' authors guarantee its indexing and processing performance for a batch operation of maximum 50,000 documents.

Tropes is indexing the documents using a preloaded dictionary. The application facilitates various adaptations and rapid enhancements of the standard dictionary in order to achieve the most profound analyzes, designed to capture features and particular aspects of the scientific discourse. Tropes dictionary is available as the script, in the form of a tree structure, offering four hierarchical levels that can be extended when needed. Tropes users can add concepts, classes of concepts and other superior structures in the attached scenario, that are embedded in the dictionary used for indexing analyzed documents. In its public version, Tropes includes an English dictionary including over 260.000 words, other options being available for French, Spanish, Portuguese, German and more recently, Romanian (ACETIC, CYBERLEX, 2013).

While indexing the scientific articles, Tropes completes three different activities (Caragea, 2011):

1) Morpho-syntactic analysis: identifies the morphological category of the words, treating simple cases of ambiguity: nouns, verbs, conjunctions, adverbs, adjectives, pronouns, articles and prepositions.
2) Lexical-semantic analysis: after processing and splitting the text into sentences, Tropes classifies the references (nouns) in semantic classes. Tropes resolves the cases of complex ambiguity by calculating the probability of occurrence of a specific word sense in a particular context, insuring an accurate identification rate of 95%.
3) Discourse analysis is an application-specific implementation, allowing an overall view on the text. Tropes provides a clear chronology of speech, the way the author introduces and manipulates references in the scene. As a result, it identifies the input and output references, cases of persistence and a relapse in the history text (Caragea, 2011).

Following the text processing and indexing stage, Tropes allows the save of semantic structures in *txt* files with the following structure:

- Files are divided into 4 sections corresponding to semantic categories: *N-4 references*, *N-3 : concepts, N-2 : classes of concepts, N-1 : categories*
- Each section contains a list of corresponding elements (see table 3), each line showing: *<concept_weight, concept_name, concept_id>*

Table 3. IDT file sample - the conceptual structure

<weight>	<name>	<id>
00000	* n-1 *// categories' section*	
00254	health	44
00137	education	120
00068	communication	21
00152	economy	54

As Tropes saves a corresponding IDT file for each processed document, its output represents only an intermediary result. The prototype read and processed the concepts from the files (using a basic text parser) and stored all the data in the *concept(i,j)* matrix.

A comparative analysis between the results of Stanford Core NLP and Tropes, showed a nearly identical output: 99% of the concepts extracted were common between the two matrices, the rest of 1% being generated by the differences between their dictionaries.

As a brief conclusion, Tropes is a valuable standalone NLP resource, especially in the case of a diverse corpus, covering six international languages and providing different dictionary features, for an optimal discourse analysis.

4. CONCEPTS WEIGHTING - TF-IDF COMPUTATION

During this phase, the raw data resulted after concepts extraction is processed for further similarity calculations. Text similarity is a largely interesting topic, with direct impact in search application, document clustering and information retrieval. The Vector Space Model (VSM) is a predominant solution for similarity computation, converting the documents in vectors of words and then measuring the distance between them (Improving text similarity measurement by critical sentence vector model, 2005). As previous studies of Raghavan (Raghavan *et al.,* 1986) and recently Lee (1997) (Lee *et al.,* 1997) proved, VSM insures a high level of accuracy to be used in large scale applications.

As a further development, in 1972, Karen Spark Jones published the initial version of the Term Frequency – Inverse Document Frequency (TF-IDF) method, computing the „importance" of a word in a corpus (Sparck Jones, 1972). The method combines the frequency of a term in a document (TF) with the ratio of that term in the whole corpus (IDF), weighting the importance of a term appearance with the number of all occurrences, giving birth to a new class of technological applications in media, language processing etc. (Ahlgrena *et al.,* 2009) As it is defined in formula 2, the *tfidf* value for a term *t*, present in document *d*, that is part of corpus *D*, is applied for each of the concepts (*concept(i,j)*).

$$tfidf(t,d) = \frac{f(t,d)}{\max\{f(w,d) : w \in d\}} \times \log \frac{|D|}{|\{d \in D : t \in D\}|} \quad (2)$$

f(t,d) is the frequency of term *t* in document *d*, *max{f(w,d): w* $\in$ *d}* is the maximum frequency of any term in document *d*, while $|d \in D : t \in D|$ is the number of documents in corpus D, containing term *t*.

This way, for each column of the *concept* matrix (the equivalent vector of frequencies for a concept) the *tfidf* can be applied. The resulting vectors are then normalized to 1 to insure a valid data input for the next phase. Concluding this phase, the input (*concept*) is translated into the *concepts_tfidf* matrix (formula 3), containing the *tdidf* values for each concept *i* in document *j*, which is now ready for similarity computing.

$$concepts\_tfidf = \begin{bmatrix} c_1 d_1 & c_2 d_1 & .. & c_M d_1 \\ c_1 d_2 & c_2 d_2 & .. & c_M d_2 \\ .. & .. & .. & .. \\ c_1 d_N & c_2 d_N & .. & c_M d_N \end{bmatrix} \quad (3)$$

5. CITATION WEIGHTING USING SIMILARITY METRICS

Finalizing the computation of TF-IDF vectors associated with each concept present in the corpus insures the context for applying the similarity metrics and calculating the *Citation Relevance Weight (CRW)* (Avram *et al.,* 2012) factors.

Therefore, for every pair of line vectors $X = (x_1, x_2, \ldots, x_n)$ şi $Y = (y_1, y_2, \ldots, y_n)$, from *concept_tfidf,* corresponding to each citation (cited article / source article), the prototype

calculated the similarity. There are three well-known solutions dedicated to the measurement of the degree of similarity between two vectors: Cosine Index, Jaccard Index, Dice Index. In a more thorough investigation, Jun Ye (Ye, 2012), has evaluated all three measures, comparing them in terms sensitivity and real life applications.

The Cosine Index (formula 4) is one of the most popular similarity measures in text processing, used in applications like clustering or data summarization. Each element of the vectors *X*, respectively *Y*, represents the terms weight in the compared documents, having a positive value, so that the results of the metric are in the [0,1] interval.

$$Cos = \frac{X \cdot Y}{\|X\|_2 \|Y\|_2} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \quad (4)$$

An important property is that cosine index is length-independent regarding the input vectors, therefore of the analysed documents (Ye, 2012). For example, the comparison of a document d_0 with a document d_1, made by appending the same content d_0 ($2 * d_0$), will return a similarity degree 1 (100%), which means that the two documents are seen as identical. So, documents having the same composition, but different totals, will be treated as being identical, which does not follow the definition of a metric, because a document made from doubling the content of another text, is a new object (Similarity Measures for Text Document Clustering, 2008). Due to the normalisation to 1 of the *tfidf* vectors, completed in the prior step (calculating TF-IDF), this disadvantage disappears in real calculation, benefiting just from the length independency.

Applying the Cosine Index for each pair of vectors, corresponding to a citing/cited article, the Citation Relevance Weight (CRW) is now obtained. Finally, using the definition of the Semantic Similarity Weight (SSW) (Avram *et al.,* 2012), in formula 5, all citation weights are normalized to their sum per each article.

$$SSW(a_i, a_j) = \frac{sem\_sim(a_i, a_j)}{\sum_z sem\_sim(a_i, a_z)} \quad (5)$$

The output of this phase is a matrix SSW, having as coordinates, both on the horizontal axis (*X*) and the vertical axis (*Y*), the lists of analysed articles, its values hosting the degrees of semantic similarity weight, calculated for each pair „*cited article X, source article Y*".

An important property of this design is that all the operations up to this phase are completely independent and can be totally isolated from the final, iterative computation. The authors have grouped them in a *preprocessing* stage, insuring that the time consuming, iterative effort is not burdened with any other operations.

6. PAGERANK ALGORITHM – COMPUTING ARTICLE BASED METRICS

Using the Article Relevance Factor (ARF) definition (Avram *et al.,* 2012), the calculation employs the PageRank algorithm; developed by the Google authors, Larry Page and Sergey Brin, the algorithm has been created using citation networks analysis and methods (Brin *et al.,* 1998).

In the matter of choosing the appropriate computing method for the citation graph, the HITS algorithm could have provided an alternative solution (Kleinberg, 1999). Even though it proved to be highly popular on the web, previous studies have found it unsuitable for bibliometrics. The reason is that a paper can receive a high score, if there are hubs citing it. As a second disadvantage, HITS results will tend to converge to zero, if the graph does not include any cycles (Sidiropoulos, 2005). Therefore, it can be concluded that HITS presents a set of particular characteristics that makes it adaptive for webpages and web-links, yet disqualifying it for computational bibliometrics.

As a successor of HITS, the Pagerank algorithm represents a meaningful enhancement of the popular number of citations, used for individual papers or diverse aggregated measures based on articles.

The citation network of scientific publications is an important resource with much more valuable information than the traditional citation counting. The PageRank is an objective measure of its citation importance that corresponds well with people's subjective idea of importance. Because of this correspondence, PageRank is an excellent way to prioritize the influence of papers, especially in research fields with a low citations pattern, therefore extracting the potential influence of scientific publications (Nan, 2008). As (Sidiropoulos, 2005) has proven, PageRank is designed in a way (which is suitable for both web and bibliometrics) that the scoring is mostly affected by the scores of the nodes that point to it and less by the number of the incoming links (citations).

Starting from the Pagerank initial definition (formula 6), the index of a web page *u*, characterized by an inbound set of links B_u, is:

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} + cE(u) \quad (6)$$

N_v is the count of outbound links on page *v* and *c* is the normalizing factor (smaller than 1), so the sum of all values *R(u)* has to be constant; the second term, *cE(u)*k, is called source of rank, as *E(u)* is a random vector with only positive values (Brin, et al., 1998).

As it is derived from Pagerank, ARF is calculated in an iterative manner. The starting values $R_0(v)$ can be randomly assigned, though the authors have used a statistical approach: $R_0(v) = 1/|| B_u ||$, meaning that all articles have the same

initial relevance factor. The loop out is only when two consecutive values are converging, so that the targeted precision is achieved. The high number of graph nodes (articles) and the previous experience with Article Influence Score (West, 2010), backed up a high precision ε, of 10^{-6}.

Based on Pagerank model (Brin *et al.,* 1998), the ARF method is described in formula 7: we consider a vector *R (relevance)* with the length of *N*, where *N* is the number of indexed articles, then $R = SSW \cdot R$. *SSW* is the matrix of semantic similarity weights, previously calculated for each pair of articles *(i, j)*, where $0 \leq i,j \leq N$.

$$\begin{aligned} &R_0 \leftarrow S \\ &loop: \\ &R_{i+1} \leftarrow SSW \cdot R_i \\ &d \leftarrow \left\|R_i\right\|_1 - \left\|R_{i+1}\right\|_1 \\ &R_{i+1} \leftarrow R_{i+1} + dE \\ &\delta \leftarrow \left\|R_{i+1} - R_i\right\|_1 \\ &while(\delta > \varepsilon) \end{aligned} \tag{7}$$

The matrix 1-norm applied for iteration *i* and *i+1*, $||R_i||_1$-$||R_{i+1}||_1$, calculates the maximum difference between the two consecutive values, while the damping factor *d* increases the rate of convergence. Matrix *R* 1-norm is calculated using formula 8.

$$\left\|R\right\|_1 = max(\sum_{k=1}^{N}\left|x_{jk}\right|), cu\ 1 \leq j \leq N \tag{8}$$

Network Loops

Following a data comparative analysis, one can observe that the graph of citations is usually simpler than a web page network, especially because of the lack of loops. Because one article can only be published once in a certain version, we face an important chronological constraint: two published articles cannot have mutual references. Situations with loop, where a paper A cites a paper B and B cites A are possible when authors exchange their working versions and cite papers not yet published, but accepted for publication. Yet these citations cannot exist in a scientific database, since the bibliographic coordinates of the published articles have changed. An obvious conclusion is that citation graphs cannot present any loops.

The most important consequence of this observation is that the damping factor *d* (also called decay-factor), initially installed because no Pagerank would ever escape from the loop and eventually the PageRank in that loop would reach infinity (PageRank as a Function of the Damping Factor, 2005), is now redundant. Furthermore, *d* can now be substituted with zero in formula 7. This particularity simplifies the calculation method, the number of iterations required for achieving convergence, and also the algorithm's processing speed.

Rank-sink

A second issue that Pagerank had to solve is the rank-sink phenomenon (loss of ranking). Generated in specific sub-networks, described only by inbound links and internal loops, the rank is artificially accumulated because of the loops, without being distributed further into the network (no external link). The reduced complexity of citation networks and the absence of loops makes it impossible for rank-sink to appear.

Dangling Links

One of the limitations of the Pagerank model is linked to the terminal nodes of the network; in the terminology proposed by Page & Brin, the links to terminal pages, which do not present outbound links, are called *dangling links*. They affect the model because they concentrate the network's ranking and their number is quite high.

In a similar way, citation networks have nodes (scientific papers) that benefit of numerous citations, but their impact on the model is significantly lowered. According to the study published by Gregory Webster in Nature, the current trend of bibliography in scientific papers is on a positive trend, the same study stating that the papers with a high number of references will equally benefit from a high number of citations (Webster *et al.,* 2009). In this context, the co-occurrence of high number of citations with high number of references is common to the majority of the current scientific papers, minimizing the disadvantages of the method.

Implementing all the above optimizations, the final solution can be described as per table 4.

Table 4. Pseudocode description for ARF computing.

Input: citation graph G, SSW matrix

Output: ARF (vector of relevance values)

```
N ← |G|
For each a ∈ G Do
  ARF_a = 1 / N
  Auxa = 0
End For // initialize vectors
While (ARF not converging) Do
   For each a ∈ G Do
    References(a) ← articles cited by a
    For each a' ∈ References(a) Do
      Aux_a' = Aux_a' + (ARF_a * SSW_aa')
    End for
  Converging = true
  For each a ∈ G Do
     converging = converging && ((Aux_a – ARF_a) < δ)
     ARF_a = Aux_a
     Aux_a = 0
  End For
End While // when ARF has converged
Return ARF
```

Algorithm performance

Considering the previous optimizations of the method, the implemented prototype has been tested on database subset containing 18000 full-text articles from social sciences and humanities. Only 3183 of them (i.e., 17.7%) have their citations stored within the corpus, providing a total of 3500 citations. The results can be characterized by the following:

- the computational effort (time) per iteration remained approximately the same, as all the SSW calculations were previously done in the preprocessing phase;
- by setting the damping factor to 0, the comparative test showed a lower number of iterations then the standard algorithm (figure 1). Figure 1 presents the number of iterations necessary for achieving convergence, while the precision was increased in 10^{-1} steps;
- excluding the damping factor determined a secondary improvement for the iteration computing effort: no multiplication between *d* and the source of rank *E(u)* is required, which for large dimensions of *E* would have induced important delays.

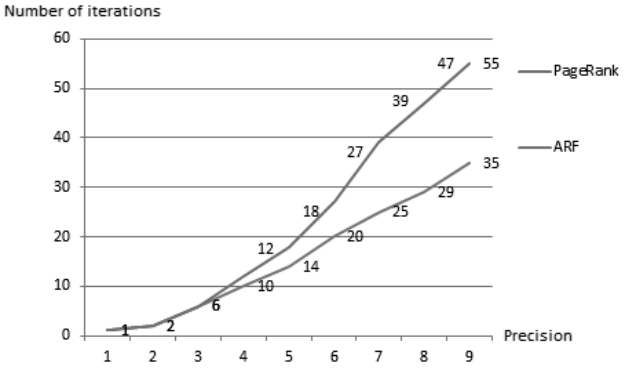

Fig. 1. Pagerank & ARF convergence rates.

7. CONCLUSIONS

This paper is addressing the practical implementation of bibliometric indicators in an attempt to meet the growing needs of academia. As article indicators have been long neglected, mainly because of a strong advertising of the journal metrics, most policy makers and financing units resumed to the existing bibliometric resources, forcing the authors and editors to realign to their new context. Yet, the answer for an adapted, more accurate bibliometric solution, capable to evaluate the real impact of research has not been provided.

If the volume of information or the paper support could have been considered obstructions for bibliometric advancements, nowadays, the technology has become the strong wave behind all information services, including bibliometrics. Search engines and other web services have proved that managing and processing *Big Data* is already a real life option.

Following the initial theoretical research, the authors have proved that a technological solution can be implemented and ran, providing fine-grained information for scientific relevance evaluation. Based on the predefined Article Relevance Factor model, critical aspects of technological implementation have been solved: conceptual similarity, citation weighting and citation network computing.

Another important achievement consists in successfully implementing and testing the natural language processing tools (Standford Core NLP and Tropes) that provided the input data for calculating the semantic similarity of the scientific discourse. This attainment offers a viable solution for citation weighting based on content analysis, disregarding inconsistent information like the number of coauthors or common keywords.

Due to the higher processing volume, compared to similar metrics like Eigenfactor and AIS, the system has been optimized by combining the semantic analysis, TF-IDF weighting and similarity computing in a preprocessing phase. Also, during the second stage of the design, graph computing algorithms have been implemented, tested and optimized for citation networks. Performance testing has shown similar iteration times, but faster convergence rates in case of the prototype, in comparison with the standard available solutions.

The results of using this metric system can ensure the informational base for evaluating the quality of scientific production, with immediate effect on:

1) evaluating the performance of the personnel involved in the research activity;
2) evaluating the performance of the research departments in universities or research institutions;
3) the cost-performance analysis, done by the research financing bodies;
4) the strategic analysis of research for local or national purposes.

Article Relevance Factor is applicable to scientific content databases that can provide detailed bibliographic information, with a relevant coverage of publications: Elsevier's Scopus and Science Direct could represent a good information source, but other aggregated databases like ProQuest Central or Cross-Check (Griffin, 2010) could also supply the data.

In terms of technology, ARF's prototype represents an integration of the state-of-the-art technologies in semantic analysis and citation network processing.

REFERENCES

ACETIC, CYBERLEX. (2013). Semantic Search Engine, Text Analysis & Semantics. *Semantic Knowledge*. http://www.semantic-knowledge.com.

Ahlgrena, P., Colliander, C. (2009). Document–document similarity approaches and science mapping: Experimental comparison of five approaches. *Journal of Informetrics*, no. 3, pp. 49-63.

Avram, S., Caragea, D., Dumitrache, I. (2012). A new approach to bibliometrics, based on semantic similarity of scientific papers. *Control Engineering and applied informatics*, Vol. 14 (3), pp. 35-42.

Ball, R., Tunger, D. (2006). Bibliometric analysis – A new business area for information professionals in libraries? *Scientometrics*, Vol. 66 (3), pp. 561–577.

Braun, T., et al. (2010). How to improve the use of metrics. *Nature*, Vol. 465 (7300), pp. 870-872.

Boldi, P., Santini, M., Vigna, S. (2005). PageRank as a Function of the Damping Factor. *Proceedings of the 14th international conference on World Wide Web,* pp. 557-566. ISBN: 1-59593-046-9. New York : ACM.

Brin, S., Page, L. (1998). *The PageRank Citation Ranking: Bringing Order to the Web.* s.l. : Stanford InfoLab, Technical Report.

Caragea, D., Badanoiu, A. et al. (2011). *Manual de autorat stiintific.*

Griffin, C. (2010). The Journal of Bone & Joint Surgery's CrossCheck experience. *Learned Publishing*, Vol. 23 (2), pp. 132-135.

Heimeriks, G., Besselaar, P.V.D. (2002). *State of the Art in Bibliometrics and Webometrics.* s.l. : Universiteit van Amsterdam.

Huang, A. 2008. *Similarity Measures for Text Document Clustering.* New Zealand Computer Science Research Student Conference, pp. 49-56.

Karlin, I. (2012). *An Evaluation of NLP Toolkits for Information Quality Assessment.* s.l. : Linnaeus University.

Kleinberg, J. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, Vol. 46(5), pp. 604–632.

Li, W. et al. (2005). Improving text similarity measurement by critical sentence vector model. *Proceedings of the Second Asia conference on Asia Information Retrieval Technology* (AIRS'05).

Lee, D.L., Chuang, H., Seafmons, K. (1997). Document Ranking and the Vector-space Model. *IEEE Software*, Vol. 14(2), pp. 67–75.

Levitt, J. M., Thelwall, M. (2011). A combined bibliometric indicator to predict article impact. *Journal of Information Processing and Management*, Vol. 47 (2), pp. 300-308.

Molette, P. (2012). Documentation sur le logiciel Tropes. *Tropes.* http://www.tropes.fr/.

Nan, M., Jiancheng, G., Yi, Z. (2008). Bringing PageRank to the citation analysis. *Journal of Information Processing and Management*, Vol. 44, pp. 800-810.

Raghavan, V.V., Wong, S.K.M. (1986). A Critical Analysis of Vector Space Model for Information Retrieval. *Journal of the American Society for Information Science*, pp. 279–287.

Ryzko, D. et al. (2011). *Emerging Inteligent Technologies in Industry.* s.l. : Springer. ISBN: 978-3-642-22731-8.

Sidiropoulos, A.., Manolopoulos, Y. (2005). A Citation-Based System to Assist Prize Awarding, *ACM SIGMOD Record*, Vol.34 (4), pp. 54-60.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, Vol. 28, pp. 11-12.

Stanford University. NLP Group. (2013). *The Stanford Natural Language Processing Group,* http://nlp.stanford.edu/ people.shtml.

The Apache Software Foundation. (2010). Apache OpenNLP. *Open NLP.* http://opennlp.apache.org/ index.html.

The Stanford Natural Language Processing Group, Core NLP Tools. (2013). http://nlp.stanford.edu/software/ corenlp.shtml.

Velter, V. (2010). ISI publications management through performance indicators. *Annals of the Constantin Brancusi University of Targu Jiu, Economy Series*, Vol. 3, pp. 119-128.

Webster, G., Jonason, P. K., Schember, T. O. (2009). Evolutionary Psychology, Vol. 7, pp. 348 – 362.

West, J.D., Bergstrom, T.C. (2010). The Eigenfactor Metrics: A Network Approach to Assessing Scholarly Journals. *College and Research Libraries*, Vol. 71 (3), pp. 236-244.

Ye, Jun. (2012). Multicriteria Group Decision-Making Method Using Vector Similarity Measures For Trapezoidal Intuitionistic Fuzzy Numbers. *Group Decis Negot*, no. 21, pp. 519-530.