WEB AUTHENTIC AND SIMILAR TEXTS DETECTION USING AR DIGITAL SIGNATURE

Marios Poulos, Nikos Skiadopoulos and George Bokos
Laboratory of Information Technology, Department of Achives and Library Sciences, Ionian University, Corfu, Greece

Keywords: Data mining, AR model, Semantic web, Information retrieval.

Abstract: In this paper, we propose a new identification technique based on an AR model with a complexity of size O(n) times in web form, with the aim of creating a unique serial number for texts and to detect authentic or similar texts. For the implementation of this purpose, we used an Autoregressive Model (AR) 15th order, and for the identification procedure, we employed the cross-correlation algorithm. Empirical investigation showed that the proposed method may be used as an accurate method for identifying same, similar, or different conceptual texts. This unique identification method for texts in combination with SCI and DOI may be the solution to many problems that the information society faces, such as plagiarism and clone detections, copyright related issues, and tracking, and also in many facets of the education process, such as lesson planning and student evaluation. The advantages of the exported serial number are obvious, and we aim to highlight them while discussing its combination with DOI. Finally, this method may be used by the information services sector and the publishing industry for standard serial-number definition identification, as a copyright management system, or both.

1 INTRODUCTION

A challenging issue rising from the phenomenon of the enormous increase of data and the requirement of data integration from multiple sources is to find near duplicate records efficiently. Near duplicate records create high similarity to each other; however, they are not bitwise-matching. There are many causes for the existence of near duplicate data: typographical errors, versioned, mirrored, or plagiarized documents, multiple representations of the same physical object, spam emails generated from the same template, etc. (Xiao et al, 2008). In recent, years many systems have been developed in order to solve the above problems. Furthermore, in the internet approach with these strongly dynamic features, many times articles are published and after a short period, they are removed from the URL location. This phenomenon many times lead to plagiarism practices. For this problem, (Phelps & Wilensky 2000) propose a less burdensome solution: compute a lexical signature for each document, or a string of about five key identifying words in the document. However, while this idea seems quite practical, this calculation is very complex and as shown, the observed complexity is achieved O(n2) times (Klein & Nelson, 2008), where n is number of the compared characters. The algorithm of the above case is dependent upon the intention of the search. In further detail, these algorithms weighted for Term Frequency (TF: "how often does this word appear in this document?") were better at finding related pages, but the exact page would not always be in the top N results. Algorithms weighted for Inverse Document Frequency (IDF: "in how many documents does this word appear?") were better at finding the exact page but were susceptible to small frequency changes in the document such as a fixed spelling (Klein & Nelson, 2008). A common statistical approach is the structure of text vectors based on values relating the text, like the frequencies of words or compression metrics (Lukashenko, et al. 2007). Based on statistical measures, each document can be described with so-called fingerprints, where n-grams are hashed and then selected to be fingerprints (Lukashenko, R., et. al 2007). In brief, the above techniques can be approximately grouped into two categories: attribute-counting systems and structure-metric systems (Chen, et al., 2004).

However, this approach causes many problems due to the pair-to-pair comparison that increases the

Poulos M., Skiadopoulos N. and Bokos G.
WEB AUTHENTIC AND SIMILAR TEXTS DETECTION USING AR DIGITAL SIGNATURE.
DOI: 10.5220/0002803600890094
In *Proceedings of the 6th International Conference on Web Information Systems and Technology (WEBIST 2010)*, page
ISBN: 978-989-674-025-2

complexity of the algorithm. Moreover, the lexical signature or fingerprint marks lead to many statistical errors. In particular, the similarity 's' measuring between documents (clone detection), a procedure known as "mental templates" (Baxter, 1998), is an example of a typical error. Briefly, these algorithms are also implemented using suffix trees, dynamic pattern matching (DPM) and hash-value comparison (Chanchal et al., 2009). In detail, the clones' software detection deal with the problems of superfluous brackets which are added in the copied fragment as it compares only the sequence of tokens and does not remove brackets before comparison.

In this study, we introduce a new approach for clone and plagiarism detection using a set of augmented linear model parameters as features for improved comparison procedure. The augmented set of linear model parameters, estimated from the document, is used as the feature vector upon which comparison procedure is based.

The objective of this paper is to extract individual-specific information from a document and to use this information in the form of appropriate features to develop a comparison clone detection method. The advantages of this method are focused on the decreased complexity of the method; in our case, this complexity is calculated in the order of O(n) times. However, the spectral linear filtering of the document data yields 15 orders' vector which is carried by all the features included the superfluous brackets and symbols. Also, we considered that the participation of these symbols in processing procedure influence the semantic interpretation of each document.

In the comparison stage, we adopted a conventional cross-correlation procedure in order to calculate the relation significance between the vectors. We called the degree of relation of these vectors the degree of similarity. Thereinafter, for accurate and verification reasons we published this algorithm in the web in particular URL location.

Finally, the proposed method was divided into the following parts:

- Pre-processing stage: the characters of a text were submitted in a conversion to numeric values and a numerically singular size array (vector) was constructed.
- Processing stage: analysis of the numeric array via a well-fitted AR model in order to extract AR coefficients.
- Comparison procedure: the degree of similarity between the investigated documents are extracted using cross-correlation technique
- Internet implementation of the above stages in Internet online testing.

2 METHOD

2.1 Pre-processing Stage

In this stage, we suppose that a selected text forms an input vector $\overrightarrow{X} = (X_1, X_2, X_3 \ldots, X_n)$, which represents the characters of the selected text. Then, using a procedure which converts a symbolic expression to American Standard Code for Information Interchange (ASCII) characters in string arithmetic values, we obtain a numerical value vector $\overline{S} = (S_1, S_2, S_3 \ldots S_n)$, with values ranging from 1 to 128. In our example, an array of characters were trialled, and we achieve this conversion by using the double.m function of the Matlab language. This function converts strings to double precision and equates itself by converting an ASCII character to its numerical representation.

For better comprehension, we provide the following example via Matlab:

```
>> S = 'This is a message to test the double "command".'
>> double(S)
ans =
 Columns 1 through 12
    84   104   105   115    32   105   115    32    97    32   109   101
 Columns 13 through 24
   115    97   103   101    32   116   111    32   116   101   115   116
 Columns 25 through 36
    32   116   104   101    32   100   111   117    98   108   101    32
 Columns 37 through 46
    34    99   111   109   109    97   110   100    34    46
```

2.2 Processing Stage

The methods employed for signal analysis and feature extraction, along with the comparison step by appropriate cross-correlation algorithm are presented in this section. The section is divided into four main subsections. In the first subsection, the choice of an AR model and the estimation of its parameters are considered. In the second subsection, "Identification Procedure" the comparison between selected pairs of texts is developed. Finally, in the

third and four subsections, the Internet Implementation and Experimental Part are described.

2.2.1 The AR Model – Type and Parameterization

In order to model the linear component of a text-file numeric conversion is implemented via a linear, rational model of the autoregressive type, AR (p), is fitted to the digitized numeric text (t) (GEP Box, et al., 1970). This signal is treated as a superposition of a signal component (deterministic) plus additive noise (random). Noise is mainly due to imperfections in the recording process. This model can be written as

$$x_t + \sum_{i=1}^{p} a_i x_{t-1} = 0 \ , \qquad (1)$$

is an independent, identically distributed driving noise process with zero mean and unknown variance σ_e^2 and model parameters $\{a_i, i = 1, 2,..., p\}$ are unknown constants with respect to time.

It should be noted that the assumption of time invariance for the model of the text vector can be satisfied by restricting the signal basis of the method to a signal "window" or "horizon" of appropriate length.

The linear model can usually serve as (more or less successful) approximations, when dealing with real world data. In the light of this understanding, the linear is the simpler among other candidate models in terms of computing spectra, covariance's, etc.;

In this work, a linear model of the specific form AR(p) is adopted. The choice of the order of the linear models is usually based on information theory criteria such as the Akaike Information Criterion (AIC) (Stone, 1977) which is given by:

$$AIC(r) = (N - M)\log\sigma_e^2 + 2r \qquad (2)$$

Where,

$$\sigma_e^2 = \frac{1}{N-M}\sum_{t=M+1}^{N} e_t^2 \qquad (3)$$

N is the length of the data record; M is the maximal order employed in the model; (N-M) is the number of data samples used for calculating the likelihood function; and r is the number of independent parameters present in the model. The optimal order r* is the minimiser of AIC(r).

We have used the AIC to determine the order of the linear part of the model in i.e. the optimal order p of the AR part of the model. For each candidate order p in a range of values [pmin, pmax], the AIC(p) was computed from the residuals of each record in the ensemble of the EEG records available. This is because we deal with recordings of real world data rather than the output of an ideal linear model. We have thus seen that AIC(p) takes on its minimum values for model orders p ranging between 10 and 15, record-dependent. In view of these findings, we have set the model order of the AR part to p = 15, for parsimony purposes

2.3 Identification Procedure

In this stage the extracted sets of the 15 order AR coefficients $\hat{C}_x$ of the vector text are submitted to the cross-correlation procedure (Morrison, et al., 1976) see equation (4).

$$r = \frac{\sum_{i=1}^{k-1}(Cx_i - \bar{C}x_i)(Cy_i - \bar{C}y_i)}{\sqrt{\sum_{i=1}^{k-1}(Cx_i - \bar{C}x_i)^2 \sum_{i=1}^{k-1}(Cy_i - \bar{C}y_i)^2}} \qquad (4)$$

In particular, the extracted cross-correlation coefficient of this procedure is a number between -1 and 1, which measures the degree to which two variable sets are linearly related. This number we adopted as a degree of similarity between the compared texts. The procedure of this comparison is described as follows according to the following algorithmic step:

1. Two texts are imported
2. The smaller extent text is selected
3. The degree of similarities (Cross correlation procedure) of the selected text is tested with equal size parts of the bigger text using a slight moving window.
4. The greater degree of similarity is selected of these multiply comparisons.

2.4 Internet Implementation-Experimental Part

2.4.1 Internet Implementation

For implementing the algorithm into a widely accessible web application, we used a Common Gateway Interface (CGI) (Robinson, et al. 2004) to interface the functions (packaged and compiled into a matlab-executable file) with the web server running on the Information Technology Laboratory. The algorithm was compiled and packaged using

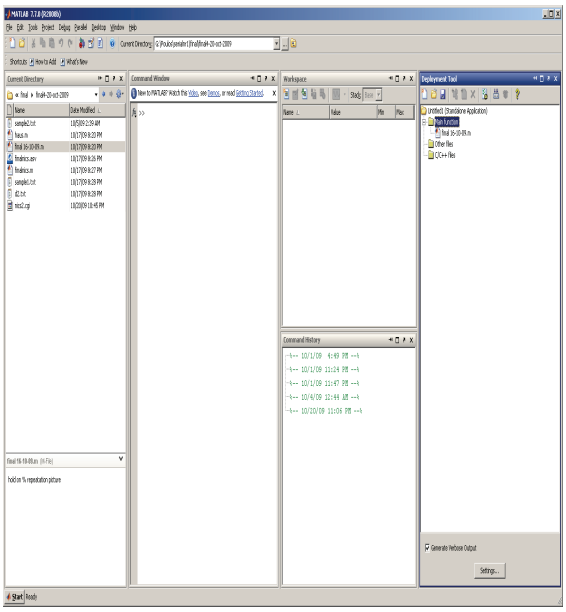

Figure 1: Using Matlab's Deployment Tool to produce a windows standalone application.

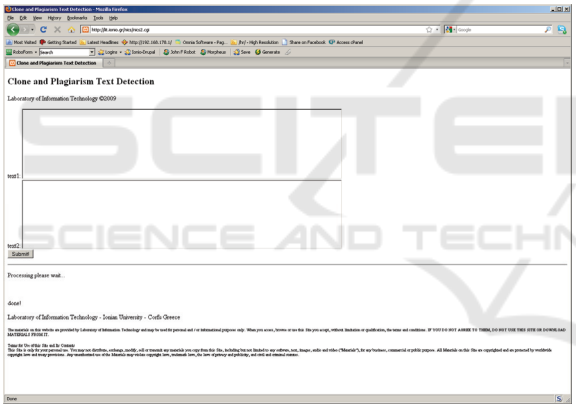

Figure 2: The Web Application.

Matlab's compiler (see figure1), producing a Windows 32-bit executable file, which was copied to the server.

The proposed algorithm was used to produce the executable file, which handles two separate text files and compares their serial numbers.

The CGI script was implemented to allow the user to input the two texts to be identified. It produces two different serial numbers, the similarity and outputs them to the user in a suitable HTML form (see figure2). Through the following Perl function, the application manages to get the two text files that the user inputs and saves them in the server's directory. Then the Perl script calls the matlab-executable file, produced earlier, and outputs the results in a suitable HTML form.

```
   use strict;
   use warnings;
   use CGI;
   my $cgi = CGI->new();
    print $cgi->header('text/html');
   print $cgi->start_html('Clone and
Plagiarism Text Detection'),
   $cgi->h2('Clone and Plagiarism Text
Detection'),
   $cgi->p ('Laboratory of Information
Technology ©2009'),
   $cgi->start_form,
   'text1: ',
   #$cgi->textfield(-size=>100,-
name=>'text'), $cgi->br,
   $cgi->textarea(-size=>10000,-
name=>'text',-COLS=>100,-ROWS=>10),
$cgi->br,
   'text2: ',$cgi->textarea(-
size=>10000,-name=>'text2',-
COLS=>100,-ROWS=>10), $cgi->br,
   $cgi->submit('Submit!'),
   $cgi->end_form, $cgi->p,
   $cgi->hr;
   open (MYFILE3, ">sample1.txt")or
die $!;
   print MYFILE3 $cgi->param('text');
   close (MYFILE3);
   open (MYFILE4, ">sample2.txt")or
die $!;
   print MYFILE4 $cgi->param('text2');
   close (MYFILE4);
   print "<p>Processing please
wait...\n</p>";
   my @a = (1);
   for my $p (@a) {
      my $pid = fork();
      if ($pid == -1) {
          die;
      } elsif ($pid == 0) {
          exec ".\\nics.exe" or die
"cannot exec program";
      }
   }
   while (wait() != -1) {}
   print $cgi->br;
   print "<p>\ndone!\n</p>";
  print "<p>\nLaboratory of
Information Technology - Ionian
University - Corfu Greece\n</p>";
```

Finally, the implementation of this application is available in the URL location http://lit.ionio.gr/nics/nics2.cgi

2.4.2 Internet Implementation

In this stage, we distinguish between three (3) different overlap measures per text:

1. **Authentication Text** - the complete overlap (degree of similarity =100%) between two equal size texts
2. **Similarity Text** - the partial overlap (degree of similarity>80%) between two equal size texts
3 **Different Texts** - the partial overlap (degree of similarity<80%) between two equal size texts.

In the first experiment, we used 50 different texts for Reuter's database and for authentication purposes, we tested a part (sentence) of these texts and submitted for the original text for example (see Figure 3)

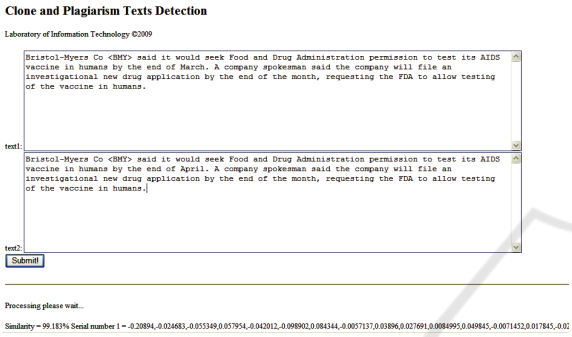

Figure 3: Comparison test between authentic parts of texts (http://lit.ionio.gr/nics/nics2.cgi).

In this way, we tested the ability of this method in order to discover the same phrase of an original text. In total, we tested 100 different cases which yielded 100% successful results.

In the second experiment, we examined the 50 texts in combination in different pairs each time. Totally, we executed 2450 tests of the above combination and all the results gave results less than 60%.

In the third experiment, we manually examined some similar texts per pair. In example, we examined texts missing a word, or similar documents differing only by typos, punctuation, or additional annotations running time. (see Figure 4).

Generally, the entire test showed that the method has specific sensitivity of the differences of words which it deals with.

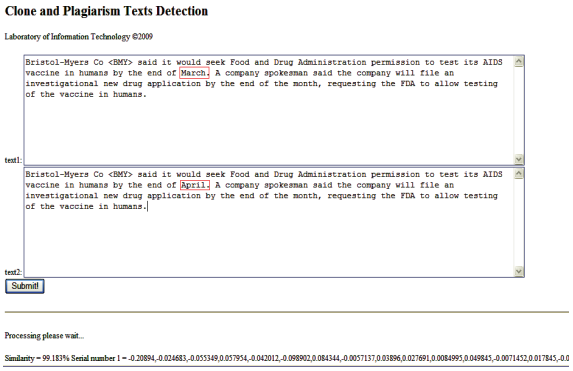

Figure 4: Comparison test between similar parts of texts (with differences in the red words) (http://lit.ionio.gr/nics/nics2.cgi).

3 CONCLUSIONS

In conclusion, by conducting this study, evidence appeared that the proposed system recognized the authentication and the degree of similarity between pair of texts in different contexts. This method may be used in many cases for the existence of near duplicate data: such as, typographical errors, versioned, mirrored, or plagiarized documents, multiple representations of the same physical object, and spam emails generated from the same template.
As a possible future application, the extracted feature vector of the proposed spectral analysis may be used as serial number for identification purposes. For example, this can be embedded into the suffix of the DOI system to enable the text retrieval capabilities through Open-URL queries. This solution was adopted because the DOI cooperates perfectly with such metadata information services as the CROSSREF and the protocol Open-URL. Specifically, it is known that an Open-URL consists of a base URL, which addresses the user's institutional link-server, and a query-string, which contains the data of this entry, typically in the form of key-value pairs. Another point of comparison that may yield guidance addressing issues in lack of similarity may be to involve the management bodies of knowledge. Future research, therefore, should be focused on further investigating the properties of the method, through experiments with large collections of documents. This way it will be possible to extensively evaluate its validity of the algorithm against a larger sample base and examine the utilization of DOI as a metadata platform-cooperation model with other information networks such as neural networks. Our long term goal is the adoption of the proposed model as a strategic component for organizations and libraries for the identification and control of the authenticity of electronically published documents on the web.

REFERENCES

Baxter, I. Yahin, A. Moura, L. & Clone Anna, M. S., 1998. Detection using abstract syntax trees. *In: Proc.* ICSM, (Intl. Conference on Software Maintenance)*Conference title. . Location. Date of Conference, Publisher: Place of publication*.

Box, G.E.P. Jenkins, G.M. & Reinsel G.C., 1970. *Time Series Analysis Forecasting and Gontrol.* Wiley John Wiley & Sons, Inc.

Chanchal K. Roya,J. R. Cordy, A. & Koschkeb, R., 2009. Comparison and evaluation of code clone detection techniques and tools: A qualitative approach. *Science of Computer Programming*, 74(#) pp.470-495.

Chen, X. Francia B., Li, M. Mckinnon, B. & Seker A., 2004. Shared information and program plagiarism detection. *IEEE Trans. Information Theory*, 7 (#),pp.1545–1550.

Klein M.& Nelson M. L., 2008. Revisiting Lexical Signatures to (Re-)Discover Web Pages. *In Proceedings of ECDL '08*, pages 371-382.

Lukashenko, R., et. al., 2007. Computer-Based Plagiarism Detection Methods and Tools: An Overview. In (*International Conference on Computer Systems and Technologies* Conference title. City,*Bulgaria 14-15 June 2007. Publisher, Place of publication.*

Morrison, N. & Donald F., 1976. *Multivariate Statistical Methods*. New York: McGraw-Hill Book Company.

Phelps, T. A.& Wilensky, R., 2000. Robust Hyperlinks: Cheap, Everywhere, Now. *In Proceedings of*
Digital Documents and Electronic Publishing 2000. (DDEP00), September 2000.

Robinson, D. &Coar K., 2004. The Common Gateway Interface (CGI) Version 1.1. RFC 3875, Oct. 2004.

Stone, M., 1977. An Asymptotic Equivalence of Choice of Model by Cross-Validation and Akaike's Criterion. *Journal of the Royal Statistical Society,* Ser. B 39, pp.44-47.

Xiao C. Wang W., Lin X. & Yu, J. X., 2008. Efficient similarity joins for near duplicate detection, *In WWW '08.*