Information Services & Use 36 (2016) 211–214
DOI 10.3233/ISU-160819
IOS Press

A methodology for digitally exploring electronic publication content

Fernando Loizides [*] and Sam Jones
Emerging Interactive Technologies Lab, Maths and Computer Science Department, University of Wolverhampton, UK
E-mails: fernando.loizides@wlv.ac.uk, SJ175@wlv.ac.uk

Abstract. The work in this article presents a methodology and coding examples to be used by those wanting to explore the content of a large corpus of digital publications stored online in PDF format and gain insight into their common content and changes over time. The method can produce hypothesis of topic popularity and point to areas for further scrutiny in terms of common emerging themes. This practitioner article provides a very specific and directed resource for information seekers, both technical in nature and not, in exploring potential themes and likely trends within a corpus.

Keywords: Electronic publishing, digital publishing, research trends

1. Introduction

The advent of the internet has brought about changes in the way that publishing takes place, both in terms of speed and cost. These changes bring about new policies to address issues that may occur; such as that of open access in academic environments [2,4]. With these changes in the ability to publish, the amount of publications, as well as the diversity of the content and publication routes has expanded drastically [1].

A common practice in order to discern trends in literature and practice is to speculate subjects, themes and directions of research through manual scanning. This includes an information seeker going over titles, abstracts and within-document texts and conceptually try to identify a subjective common theme. This process takes considerable time and effort; often more than is available to the information seeker or even a group. The process also has methodological limitations such as the inability to identify themes based on the frequency of occurrence of different words in the text. These can be overlooked from abstracts and titles. In this work we produce a simplified method for identifying numerical frequency of occurring words across several documents. The underlying principle is that if a term appears in several separate documents, then the likelihood that this is a common theme is elevated.

2. Methodology

Electronic publications (the files) are often stored in PDFs (Portable File Format) which is 'readable' by a variety of non-proprietary, free and open source software. "These combine text and graphics by

[*] Corresponding author. E-mail: fernando.loizides@wlv.ac.uk.

treating the glyphs that express text as little pictures in their own right, and allowing them to be described, denoted, and placed on an electronic "page" alongside conventional illustrations. They portray finished documents, ones that are not intended to be edited, and are therefore more akin to traditional library documents than word-processor formats" [7].

For our methodology, we will focus on finding, extracting and using files from this format, although this can be adjusted to accommodate other formats such as Hypertext Markup Language (HTML) files (webpages) or plain text files. The methodology and code we present is both specific in the formats it uses but also generic in that a minimal coding skill is adequate to alter the constraints and potential of use to the users' needs.

Our process includes three stages. The first stage includes scraping (searching through) websites in order to extract the PDFs of the publications. The second stage includes converting the files from PDF to plain text. The third and final stage includes analyzing the files for content. We will now cover these three stages using pseudocode type algorithm to outline and explain the high level design and internal workings. For a more detailed view of the exact code please see the Appendix.

2.1. Stage 1: PDF scraping

The first stage involves identifying and finding the documents (PDFs) and extracting them from the websites/repositories they reside in. The steps to do this are as follows:

1. Declare the website location (URL)
2. Find all the links that link to the different years of publication/grouping that you require. This may require some adapting to get the categorization right. You may need to adapt the webpage slightly if the structure of the web pages is vague.
3. Find the appropriate element class that describes the PDF (or the file type you require) and associate that link to what needs to be downloaded.
4. Go through the links and download files.

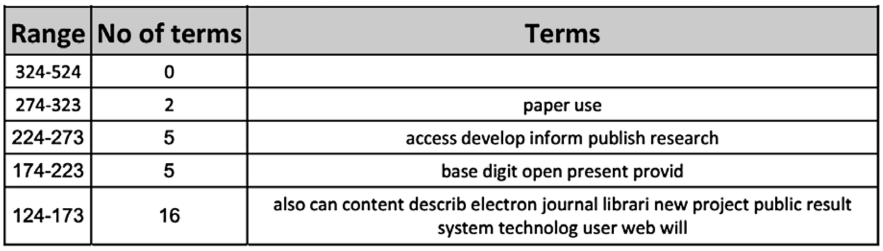
For the detailed code written in Python, see Appendix A.

2.2. Stage 2: Converting the files to text

The second stage requires us to change the files we have scraped (extracted from the web pages) into plain text. For our needs, we do not add weighting. Weighting is a term used to indicate importance of a word or phrase compared to its location. For example, if a word occurs in a title, then we can assume that it is more likely to be more 'import' than if it occurs in the main plain text of a document. The pseudocode for this is as follows:

1. Load the required tools to convert the types of files that you need. We use PDF and MS Word file conversions.
2. Load the saved files/database of files that were extracted in Stage 1
3. Decide if the file type is a PDF or an MS Word document (or any other type).
4. Go through the individual pages and extract each word and write it to a separate text file.

The code, written in Python to do this can be found in Appendix B.



Range	No of terms	Terms
324-524	0	
274-323	2	paper use
224-273	5	access develop inform publish research
174-223	5	base digit open present provid
124-173	16	also can content describ electron journal librari new project public result system technolog user web will

Fig. 1. Word analysis table.

2.3. *Stage 3: Text mining of the plain text documents*

The final third stage includes the

1. Remove common words (e.g. ‘the’ ‘and’)
2. Remove Specific Characters (e.g. ‘-’ ‘.’ ‘∗’ and punctuation marks)
3. Transform all text to lowercase (e.g. INFORMATION to information)
4. Remove Digits (it was deemed appropriate for our research that digits would not contribute to the findings)
5. Strip excess whitespace
6. Match Spelling (change all American to English or vice-versa)
7. Apply Porter Stemming Algorithm [5,6] (Stemming reduces words to their most basic state in order to identify similar words – e.g. visualize and visualizing would become visual).
8. Count the individual words based on year (or grouping of your choosing).
9. Present the most popular words in terms of the occurrence in the most different categories.

The code for this can be found in Appendix C.[1] The cleaning process took place using R (https://www.r-project.org). Once the cleaning process takes place, the documents were queried (again using R) as to the cumulative term frequency of each word across the documents (full texts and abstracts). The results coming from this code can produce results such as the ones presented from a study performed on the Electronic Publishing Conference Digital Library [3] (see Fig. 1). From these, the user can then distinguish words that occur across different amounts of documents, the most occurring being the most likely to be trending. The information seeker can also choose to classify the different results in years, or groups. In [3] the authors also experimented with abstracts only and compared different time periods.

3. Conclusions

In this work we present a methodology and technical information in terms of coding examples for the discovery of potential themes or topics within a corpus by investigating word frequency at a more micro analysis level. The methodology provides a foundation that can be adapted to the needs of the information seeker rather than a rigid step by step algorithm. The aim is for code literate and non-code literate audiences to be able to use the methodology in order to explore documents or assist in describing how to create a searching strategy to a developer.

[1] Code based on an adaptation from that found online from: https://rstudio-pubs-static.s3.amazonaws.com/31867_8236987cf0a8444e962ccd2aec46d9c3.html.

There are limitations in our method. For example, from a technical perspective, there may be texts which are photocopies and scans which require other techniques to extract the text (see Optical Character Recognition). Another limitation is when there is no access to some of the texts. There is also no phrase matching. A possible direction for further work would be to extract some sort of contextual data from the files. Much of the analysis in this paper is based around the frequency of which words appear but if one were able to extract some contextual data it may be that more insightful inferences could be made. This is likely to be a difficult problem involving complicated NLP and text mining techniques.

Supplementary data

Appendix is available at: http://dx.doi.org/10.3233/ISU-160819.

References

[1] L. Bornmann and R. Mutz, Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references, *Journal of the Association for Information Science and Technology* **66**(11) (2015), 2215–2222. doi:10.1002/asi.23329.

[2] S. Harnad and T. Brody, Comparing the impact of open access (OA) vs. non-OA articles in the same journals, *D-lib Magazine* **10**(6) (2004).

[3] F. Loizides and S.A. Jones, Insights from over a decade of electronic publishing research, in: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, p. 119.

[4] J. Parker and E. van Teijlingen, The Research Excellence Framework (REF): Assessing the impact of social work research on society, *Practice* **24**(1) (2012), 41–52. doi:10.1080/09503153.2011.647682.

[5] M.F. Porter, An algorithm for suffix stripping, *Program* **14**(3) (1980), 130–137. doi:10.1108/eb046814.

[6] P. Willett, The Porter stemming algorithm: Then and now, *Program* **40**(3) (2006), 219–223. doi:10.1108/00330330610681295.

[7] I.H. Witten, D. Bainbridge, G. Paynter and S. Boddie, Importing documents and metadata into digital libraries: Requirements analysis and an extensible architecture, in: *International Conference on Theory and Practice of Digital Libraries*, Springer, Berlin Heidelberg, 2002, pp. 390–405.