Automated Information Extraction from Empirical Software Engineering Literature: Is that possible?

Daniela Cruzes[1,2],
Manoel Mendonça[1]
[1]NUPERC/UNIFACS,
Salvador, BA- Brazil
[2]FEEC/UNICAMP
Campinas, SP- Brazil
{daniela, mgmn}
@unifacs.br

Victor Basili
Dept. of Computer
Science, University
of Maryland,
College Park, MD,
20742, USA
basili@cs.umd.edu

Forrest Shull
Fraunhofer Center -
Maryland,
4321 Hartwick Road,
College Park, MD,
20740, USA
fshull@fc-md.umd.edu

Mario Jino
FEEC/UNICAMP
Caixa Postal 6101
13083-970
Campinas (SP), Brazil
Jino
@dca.fee.unicamp.br

Abstract

The number of scientific publications is constantly increasing, and the results published on Empirical Software Engineering are growing even faster. Some software engineering publishers have begun to collaborate with research groups to make available repositories of software engineering empirical data. However, these initiatives are limited due to data ownership and privacy issues. As a result, many researchers in the area have adopted systematic reviews as a mean to extract empirical evidence from published material. Systematic reviews are labor intensive and costly. In this paper, we argue that the use of Information Extraction Tools can support systematic reviews and significantly speed up the creation of repositories of SE empirical evidence.

1. Introduction

The number of scientific publications is continuously increasing, and the number of journals reporting on results from Empirical Software Engineering is also growing. In this scenario, it is important to have approaches to execute secondary studies, i.e., studies that draw conclusions over the evidence collected from previous studies.

Systematic Review [3] is quickly becoming the approach of choice to integrate evidence from Software Engineering literature. The systematic review process requires that a user identify a comprehensive collection of articles, extract information from those articles, verify the accuracy of those extracted facts, and analyze the extracted facts using either qualitative or quantitative techniques. Although a systematic review accurately captures evidence, the process is costly, taking several months from conception to publication [2] and many hours of effort [7].

Therefore, it is unquestionnable that the area would profit from tools and methods that could help to locate, organize, and summarize information for systematic reviews, as well as to synthesize it into usable knowledge [4]. The question one should ask is: Can such tools be built? This paper investigates the use of Text Mining to accomplish some of these tasks. In particular, it focuses on the use of Information Extraction Techniques to locate and organize information in documents for systematic reviews.

Text Mining (TM) is about looking for patterns in natural language text [14]. It recognizes that complete understanding of natural language text is not attainable and focuses on extracting small pieces of information from text with high reliability.

Information Extraction is a technique used to detect relevant information in larger documents and present it in a structured format. It is used to analyze the text and locate specific pieces of information in it [10].

Information Extraction (IE) is one of the most prominent techniques currently used in TM. It is a starting point to analyze unstructured text. In particular, by combining Natural Language Processing (NLP) tools, lexical resources, and semantic constraints, it can provide effective modules for mining documents of various domains [10]. Peshkin and Pfeffer [11] define IE as the task of filling template information from previously unseen text which belongs to a pre-defined domain. Its goal is to extract from documents salient facts about pre-specified types of events, entities, or relationships. These facts are then entered automatically into a database, which may then be used for further processing.

Although this approach has been used for systematic reviews in other fields [4], empirical software engineering researchers extract information manually [8][15]. There are no specific tools for the area. It is our position that information extraction techniques can significantly help the area if customized tool are made available for SE researchers.

This paper discusses this issue as follows. Section 2 introduces the basics of Information Extraction. Section 3 discusses the application of a general purpose IE tool to SE literature. Section 4 lists our conclusions and plans for future research.

2. Background: Information Extraction.

Research and development concerning Information Extraction have picked up in the late 80s. Research has been focused through the Message Understanding Conferences (MUC) [16], which has focused on the definition and evaluation of IE systems.

Information Extraction techniques can be applied to structured, semi-structured, and unstructured texts. For the latter one, Natural Language Processing is necessary and has to be combined with traditional Information Extraction systems.

An IE system: 1) identifies and 2) extracts specific information located in non-structured textual data, and 3) generates the output as has been requested. IE systems are domain specific because they extract particular entities or events from a particular domain skipping over the irrelevant ones. The kind of information to extract consists in a pre-specified set of entities and their attributes, as well as relationships and events relating those entities.

Information Extraction (IE) concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from unstructured text. Some authors provide a survey on general purpose information extraction systems, as well as a summary of evolution of the field [9] [10].

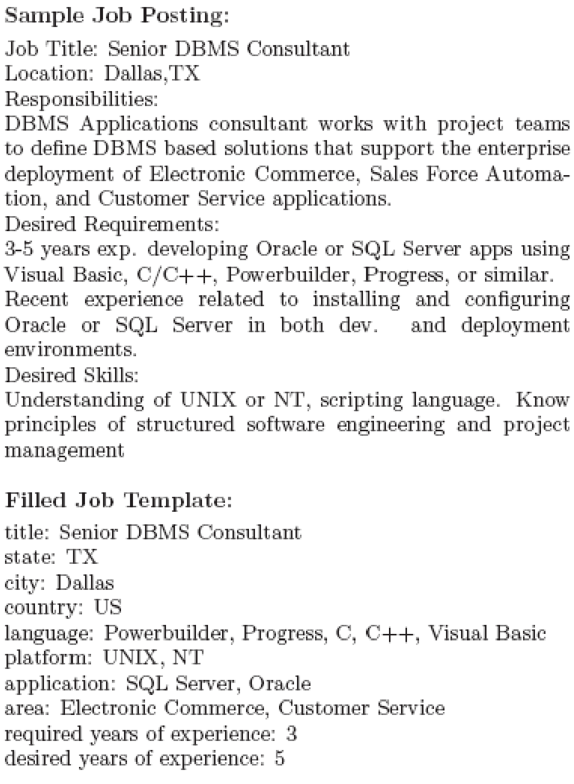
The biomedical area is one in which the use of text mining to explore scientific literature is becoming increasingly important. In recent years, several different systems have been developed. Some aim at detecting interactions among proteins, genes or both. Others specifically detect protein and gene names. Others, more specialized, extract information relating to gene expression profiling, drugs and genes relevant to cancer, signal-transduction pathways and associated drugs and diseases and c-DNA clones [17].

We identify three main types of IE techniques that can help with our goal of automatically locating and organizaing information for systematic reviews.

The first type, named entity recognition, involves identifying references for particular kinds of objects such as names of people, companies, and locations [6].

The second type of technique aims at extracting relations between entities of interest. In biomedical texts, this approach was used to identify that certain proteins interacts with other proteins or that a given protein was located in a particular part of the cell [12].

The third type of technique aims at extracting fillers for a predetermined set of slots in a particular template relevant to a certain domain [17]. Califf and Mooney, consider the task of extracting a database from postings to the USENET newsgroup, austin.jobs [13] Figure 1 from [13] shows a sample message from the newsgroup and the filled computer-science job template where several slots may have multiple fillers. For example, slots such as languages, platforms, applications, and areas usually have more than one filler, while slots related to the job's title or location usually have only one filler.



Sample Job Posting:

Job Title: Senior DBMS Consultant
Location: Dallas,TX
Responsibilities:
DBMS Applications consultant works with project teams to define DBMS based solutions that support the enterprise deployment of Electronic Commerce, Sales Force Automation, and Customer Service applications.
Desired Requirements:
3-5 years exp. developing Oracle or SQL Server apps using Visual Basic, C/C++, Powerbuilder, Progress, or similar. Recent experience related to installing and configuring Oracle or SQL Server in both dev. and deployment environments.
Desired Skills:
Understanding of UNIX or NT, scripting language. Know principles of structured software engineering and project management

Filled Job Template:

title: Senior DBMS Consultant
state: TX
city: Dallas
country: US
language: Powerbuilder, Progress, C, C++, Visual Basic
platform: UNIX, NT
application: SQL Server, Oracle
area: Electronic Commerce, Customer Service
required years of experience: 3
desired years of experience: 5

Figure 1 - Sample Job Posting and Filled Template [13]

3. Applying a IE tool to SE Literature

As a feasibility study of the use of automated information extraction tools in Software Engineering we ran an entity recognition tool on 9 papers used on a previous Systematic Review [15]. The chosen tool was

Site Content Analyzer[1]. It can examine textual or HTML documents to provide a detailed report about its word density, frequency, their weight and relevance. Also, the program offers customizable sorting filters to cut off irrelevant words or phrases.

We run the tool over the 9 pre-studied papers without any customization for SE vocabulary. We configured the tool to report up to 20 keywords for each paper. We then rated the words as either "useful" or "not useful" as a keyword. Examples of "not useful" words that were found are: data, usage, subject, IEEE, etc. Even without a customization to avoid capturing such words, the tool found around 65% of useful key words in papers Pi (P1: 50; P2: 30; P3: 50; P4: 90; P5:80; P6:80; P7: 50; P8: 70; P9: 90). Combined with clustering algorithms this result can used to automatically group documents in a systematic review [5].

We also found that correlated articles issued similar word frequency rankings, indicating that the tool can be used to compare documents. We also observed that there is a strong relationship between word frequency and document title, pointing to the usefulness of this metric to select articles for systematic reviews. In some situation, this metric produced even more specific descriptions than the paper title itself.

4. Conclusions and Future Research.

Text mining has been successfully used in fields such as biology and medicine [1][9]. Our initial investigation shows that it is also promising for software engineering. We are currently working on adapting IE techniques to the empirical SE area. In particular, we are working under the framework of a previously developed methodology to extract SE knowledge from papers [8][5].

Although adapting an IE tecnique to new topics is an expensive process, requiring both IE and domain knowledge, our position is that this effort is very worthwhile both from the perspective of information extraction speed up and repeatability.

5. References

[1] A. Divoli and T.K. Attwood. BioIE: Extracting Informative Sentences from the Biomedical Literature, Bioinformatics, 21: 2138-2139, 2005.

[2] A. Petrosino. Lead Authors of Cochrane Reviews: Survey Results. Report to the Campbell Collaboration, University of Pennsylvania, Cambridge, MA, 1999.

[3] B. Kitchenham. Procedures for undertaking systematic reviews. Technical Report TR/SE-0401, Department of Computer Science, Keele University, Australia Ltd, 2004.

[4] C. Blake. Information Synthesis: A New Approach to Explore Secondary Information in Scientific Literature. Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries, pp 56-64, Denver, CO, US, 2005.

[5] D. Cruzes, M. Mendonca, V. Basili, F. Shull and M. Jino. Using Context Distance Measurement to Analyze Results across Studies. ESEM 2007, Madrid, 2007.

[6] D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. Machine Learning, 34:211–232, 1999.

[7] E. Allen and I. Olkin. Estimating Time to Conduct a Meta-analysis from Number of Citations Retrieved. Journal of American Medical Association, 282(7), 634-635, 1999.

[8] F. Shull, D. Cruzes, V. Basili and M. Mendonca. Simulating Families of Studies to Build Confidence in Defect Hypotheses. Journal of Information and Software Technology, vol. 47(15): 1019-1032, December, 2005.

[9] H. Al-Mubaid. A Text-Mining Technique for Literature Profiling and Information Extraction from Biomedical Literature, ISSO Y2005 Annual Report , p. 45-49, 2005.

[10] K. Kaiser and S. Miksch. Information Extraction: A Survey. Vienna University of Technology, Institute of Software Technology and Interactive Systems, Vienna, Technical Report, Asgaard-TR-2005-6, May 2005.

[11] L. Peshkin and A. Pfeffer. Bayesian information extraction network. In Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI), 2003.

[12] M. Craven and J. Kumlien. Constructing biological knowledge bases by extracting information from text sources. In Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB), p 77–86, Heidelberg, Germany, 1999.

[13] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), p 328–334, Orlando, FL, July 1999.

[14] M. Hearst. What is text mining. http://www.ischool.berkeley.edu/~hearst/text-mining.html, 2004.

[15] N. Juristo, A. M. Moreno, S. Vegas. Reviewing 25 years of testing technique experiments. Empirical Software Engineering, 9: 7-44, 2004.

[16] Ralph Grishman and Beth Sundheim: Message Understanding Conference: A Brief History. Proceedings of the 16th International Conference on Computational Linguistics (COLING), 466– 471 Kopenhagen, 1996.

[17] Raymond J. Mooney and R. Bunescu. Mining knowledge from text using information extraction, ACM SIGKDD Explorations, v.7 n.1, p.3-10, June, 2005.

[1] http://www.sitecontentanalyzer.com/