Profiling and Tracing Stakeholder Needs

Pete Sawyer, Ricardo Gacitua, Andrew Stone

Lancaster University, Lancaster, UK. LA1 4WA
{sawyer, gacitur1}@comp.lancs.ac.uk, a.stone1@lancs.ac.uk

Abstract. The first stage in transitioning from stakeholders' needs to formal designs is the synthesis of user requirements from information elicited from the stakeholders. In this paper we show how shallow natural language techniques can be used to assist analysis of the elicited information and so inform the synthesis of the user requirements. We also show how related techniques can be used for the subsequent management of use requirements and even help detect the absence of requirements' motivation by detecting unprovenanced requirements.

Introduction

Since "the majority of requirements are given in natural language, either written or orally expressed" [1] the application of natural language processing (NLP) to Requirements Engineering (RE) has been investigated by many researchers. In this short paper we discuss the use of *shallow* NLP techniques in the early stages of transitioning from stakeholders' need to formal designs; the synthesis of user requirements that are informed by information elicited from the stakeholders and the subsequent management of this information. We also consider the conundrum posed by missing or suppressed information and the perhaps paradoxical potential for shallow techniques to detect the absence of information.

Assisting the Synthesis of User Requirements

Among the most challenging applications of NLP in RE have been to problems where the language used is uncontrolled [2]. Uncontrolled language is characteristic of early-phase RE [3] where the stakeholders not only hold different perspectives on the problem domain but express their needs in ways that often fail to conform to conventions of language use. The three bloggers in the airport security case study [4] illustrate this well. Even ignoring the divergence of semantics and pragmatics of their perspectives on the problem, a number of lexical and syntactic characteristics of the text pose real natural language processing problems, such as idioms ('come on!'),

implicit context ('we can deal with it.') and grammatical errors and typos ('We have to ban on ..', 'Channel No. 5').

The characteristics illustrated by the airport security blog illustrate why the automatic synthesis of user requirements is way beyond the current state-of-the-art. Useful support can be provided, however. A number of researchers have investigated the identification of domain concepts by analysis of the text using, for example, frequency profiling [5] and lexical affinities [6]. Such work can serve to help identify entities in the problem domain and their relationships, reveal key terms and populate glossaries. For example, in the airport security blog, the left hand pair of columns in table 1 shows a ranked list of the ten words and their parts of speech that occur with a frequency that most exceed the frequency predicted by the 100 million word British national corpus (BNC). Note for instance that even though "oxidizer" appears only twice (once in singular and once in plural form) in the 603 word blog, twice is still significantly more frequently than predicted by its rate of occurrence in the BNC.

There are several interesting things to note here. The first is that "screen". "screening" and "screener" all share the same word stem so could have been collapsed into a single term. That hasn't been done because in the blog they represent sufficiently different concepts to make it worth distinguishing between them. Note that "screen" is a verb while the other two terms are nouns. Even "screener" and "screening", which are both nouns, are distinguished by the different semantic tags assigned by the tool we used to generate the data (Wmatrix [2]). "screening" is classified using the semantic tag A10 *Open/closed; Hiding/Hidden; Finding; Showing*. "screener" is classified as Z99 *Unmatched*. In other words, the semantic tagger failed to recognize "screener". Interestingly, there are six occurrences of "screening" in the text. Four are nouns and two are verbs. The verb form of "screening" is not as over-represented as the noun form so it does not appear in the top ten.

The two occurrences of "oxidizer" causing it to appear as the sixth most over-represented term illustrates why the application of statistical techniques to small volumes of text tends to yield results that should be interpreted with caution. The fact that a single blogger mentioned the term twice does not *per se* mean that it represents a significant concept within the bloggers' universe of discourse. That it *might* be significant can only be determined by a skilled analyst.

The third and fourth columns in table 1 show the same as the first and second columns but this time, instead of restricting our analysis to the blog, we have included a small corpus of documents containing approximately 8000 words. This corpus was compiled quickly from a mixture of press reports about airport security and advice on security published on travel websites. Hence, we cannot claim that it is truly representative of the domain. However, it is interesting to compare the first and third columns to help understand the focus of the blog within the general domain of airport security. If we had more confidence in the relevance and degree of consensus represented by the the corpus, we could use the results of the analysis as the starting point for the construction of a domain ontology that could be used for the reuse of knowledge across airport security applications. Given the degree of uncertainty over the veracity of our hastily-compiled corpus, the most we can claim in this instance is that it reveals some of the general context of the bloggers' conversation.

Table 1. The 10 Most Over-Represented Words in the blog and a domain corpus

Blog		**Blog + domain corpus**		**Blog + domain corpus**
Term	**PoS**	**Term**	**PoS**	**Verb**
screener	noun	airport	noun	access
security	noun	security	noun	screen
airport	noun	passenger	noun	check
administration	noun	new	adj	profile
government	noun	travel	noun	travel
oxidizer	noun	system	noun	identify
screening	noun	capta	noun	Travele *(sic)*
screen	verb	surveillance	noun	carry
tsa	noun	flight	noun	allow
ban	verb	luggage	noun	capture

The fifth column of table 1 is also a ranked list extracted from the blog and the corpus. This time, however, we have filtered it on verbs to show only the action words. While the other lists are predominantly nouns it is interesting to note that "screen" reappears as a significantly over-represented action, as do "check", "profile", "identify" and other words related to the active application of security checks in airports. Sawyer et al. [2] elaborate the use of statistical techniques for analyzing elicited requirements information and advocate combining a set of different shallow NLP techniques to provide a number of perspectives on the information embodied in requirements text.

Upstream Trace Recovery

The process of user requirements synthesis is the first step in transitioning from the informal to the formal, although it is far from a simple activity and may involve (for example) goal modeling, scenario derivation, brainstorming and much else. Given the complexity of the process, it is good practice to record the synthesized requirements' motivation since maintenance of an explicit record helps inform trade-offs and allows backwards tracing to the stakeholders or information sources that motivated the requirements. Such upstream or *pre-requirements specification tracing* (pre-RST) [7] is, for a variety of reasons, commonly neglected.

Down-stream tracing (post-RST) is also commonly neglected, despite the ready availability of commercial requirements management (RM) tools that directly support post-RST. This failure of basic RM practice has motivated several researchers to investigate automatic post-RST recovery. Techniques borrowed from information retrieval (IR) have been shown to be capable of inferring relationships between requirements at different levels of elaboration [8, 9]. We have applied similar techniques to pre-RST using our *Prospect* tool [10].

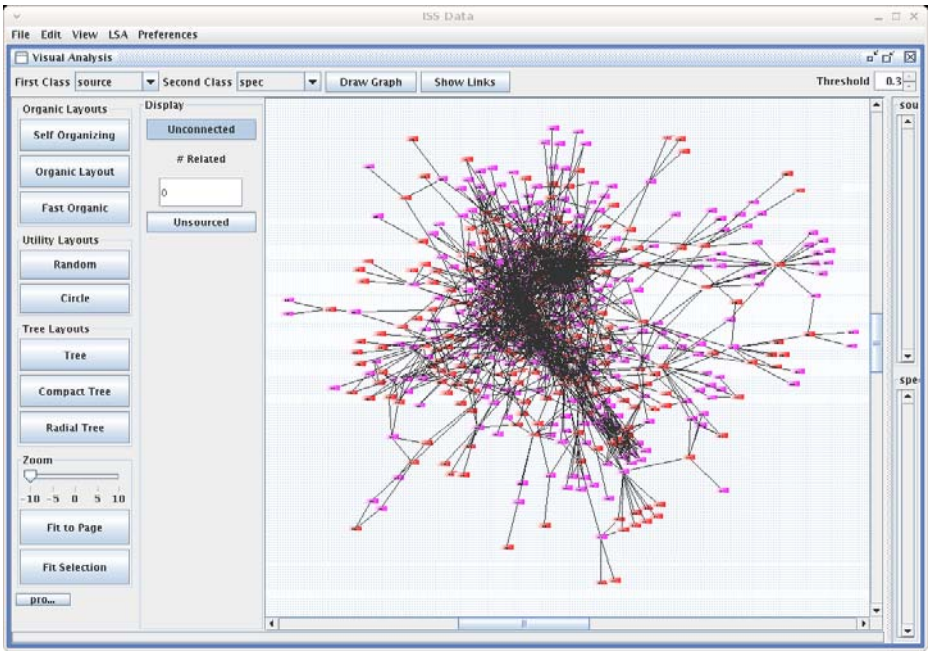

Figure 1. An Organic Layout Algorithm Used to Display Pre-RST Derives Relationships

The results of our evaluations indicate that the IR technique at the core of Prospect, latent semantic analysis (LSA) [11], is capable of inferring *derives* relationships between user requirements and the elicited knowledge that motivated them. In other words, a user requirement that participates in a relationship with passages of elicited text is motivated in some way by the information embodied by the elicited text. Fig 1. shows a cluster of several hundred requirements and passages of text from the information elicited form the stakeholders in a project. The scale is too small to see clearly here but requirements and "source" passages are represented as nodes with different colours. The arcs represent derives relationships and the tight clusters reveal where the multiplicity of relationships is high.

Although LSA is a shallow technique, it is computationally intensive. Its use is justified by its unique property of tolerating inconsistent vocabulary. LSA can infer a relationship between (say) a user requirement and passages of elicited information even when the terms used are dissimilar, provided that the terms that are used occur sufficiently commonly in similar contexts for LSA to infer synonymy or polysemy. Hence for example, if "airline" and "carrier" were used synonymously by different stakeholders, we would want pre-RST recovery to treat them as the same concept.

In practice, Prospect achieves a level of recall and precision broadly consistent with the figures shown in Fig 2 which are derived from one of our case studies. Note that the "Threshold" value that calibrates the X axis represents the tool's adjustable sensitivity. The higher the threshold of similarlty, the better the precision (i.e. fewer false positives) but the lower the recall (i.e. more valid relationships are missed). The

technique will always produce false positives, but our experiments with user groups suggest that analysts can tolerate surprisingly high levels of imprecision in return for high recall.

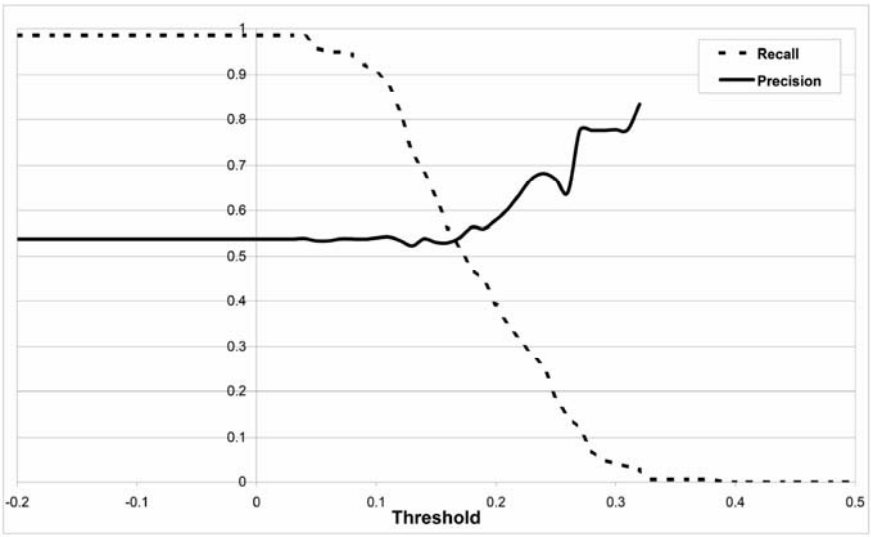

Figure 2. Recall and Precision Achieved by *Prospect*

Unprovenanced Requirements

An interesting phenomenon that our pre-RST tool commonly reveals is that of *unprovenanced* requirements. If the elicited information exists in text form, Prospect if typically able to infer derives relationships between user requirements and passages of the elicited text. The strength of a relationship between a user requirement and passages of elicited text can vary according to the lexical similarities that exist between them, but with the tolerance of synonymy and polysemy that LSA affords. In the case studies conducted so far, a minority of requirements appear to have no relationship with the elicited text. The largest of our case studies was conducted on a live project and we were able to interview the analysts to validate the results. Their responses showed a strong correlation between requirements identified by Prospect as unprovenanced and those where the requirements had been "invented" by application of the analysts' domain knowledge.

Clearly, invention is part of the job of an analyst because they must use their knowledge and experience to creatively add value to the needs stated by the stakeholders. One common reason for the need for invention is that the information elicited from the stakeholders is incomplete. Incompleteness can be due to a number of reasons, but one is that the stakeholders hold information that they don't articulate either through deliberately withholding it or (we assume, more commonly)

unconsciously withholding it. Knowledge that is never articulated, either because it is hard to articulate, or is so integral to the holder's model of the world that they don't feel the need to make it explicit is *tacit* [12, 13, 14].

A number of elicitation methods exist that help cope with tacit knowledge or concealed information [15]. EasyWinWin [16], for example, is designed to identify, refine and reach consensus on the requirements for a system over a series of steps. These steps are carefully structured using prompts and the staged revelation of stakeholders' requirements and priorities to tease out concealed information. We hypothesize that techniques such as LSA could enhance tool support for such methods by, for example, tracing the evolution of stakeholders' requirements over stages in the elicitation process and helping highlight discontinuities that might be revealing of concealed information or tacit knowledge. Hence, in addition to helping detect the effect of tacit knowledge in sets of requirements, LSA may be useful in drawing tacit knowledge and concealed information out of stakeholders during requirements elicitation.

Conclusions

In [17], Kevin Ryan offered a critique of the application of natural language processing techniques to requirements engineering problems. Among Ryan's key observations was that it was both unfeasible and undesirable to automate the derivation of requirements from natural language text. Fourteen years later, Ryan's view still holds. Instead, work has focused on using NLP techniques as a tool to aid the human analyst. We argue that in the early stages of RE where the language is inevitably uncontrolled, shallow NLP techniques hold real promise as the basis for viable analysts' tools.

One of the reasons why the automation of the analyst's task is unfeasible and undesirable is that much of the information that the analyst needs in order to formulate appropriate requirements is likely to be unstated. We have described how latent semantic analysis, when applied to up-stream trace recovery can highlight disconnects between the formulated requirements and the information elicited from stakeholders. It appears that this disconnect is sometimes a symptom of missing or incomplete information, which in turn can be caused by stakeholders failing to articulate their knowledge. We believe that the ability to detect evidence of tacit knowledge is useful in itself and may form a component in a toolset for improving how tacit knowledge is handled within RE.

References

[1] http://fabrice.kordon.free.fr/Monterey2007/home.html

[2] Sawyer, P., Rayson, P., Cosh, K.: "Shallow Knowledge as an Aid to Deep Understanding in Early-Phase Requirements Engineering", IEEE Transactions on Software Engineering, 31 (11), November 2005.

[3] Yu, E. (1997) “Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering”, Proc. Third IEEE International Symposium on Requirements Engineering (RE’97), Annapolis, MD. USA.
[4] http://fabrice.kordon.free.fr/Monterey2007/invitation_files/case-1.pdf
[5] Lecœuche, R. (2000) “Finding comparatively important concepts between texts”, Proc. Fifteenth IEEE International Conference on Automated Software Engineering (ASE’00), Grenoble, France.
[6] Maarek, Y. and Berry, D. (1989) “The Use of Lexical Affinities in Requirements Extraction”, Proc. fifth International Workshop on Software Specifications and Design, Pittsburg, Pa, USA
[7] Gotel, O., Finkelstein, A.: “An analysis of the requirements traceability problem“, Proc. 1st International Conference on Requirements Engineering (ICRE’94), Colorado Springs, Co., USA, April, 1994.
[8] Natt och Dag, J., Regnell, B., Carlshamre, P., Andersson, M., Karlsson, J.: “A Feasibility Study of Automated Support for Similarity Analysis of Natural Language Requirements in Market-Driven Development”, Requirements Engineering, 7 (1), 2002.
[9] Huffman-Hayes, J., Dekhtyar, A., Karthikeyan Sundaram, S.: “Advancing Candidate Link Generation for Requirements Tracing: The Study of Methods”, IEEE Transactions on Software Engineering, 32 (1), January 2006.
[10] Stone, A., Sawyer, P.: “Identifying Tacit Knowledge-Based Requirements”, IEE Proceedings Software, 153 (6), December 2006.
[11] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: „Indexing by latents semantic analysis“, J. Am. Soc. For Inf. Sci., 41 (6), 1990.
[12] Polanyi, M.: The Tacit Dimension, Peter Smith, Gloucester, Ma. USA, 1983.
[13] Nonaja, I, Takeuchi, H.: “A theory of organizational knowledge creation”, Int. J. of Technology Management, 11 (7/8), 1996.
[14] Busch, P. and Richards, D.: “Acquisition of Articulable Tacit Knowledge”, In Proc. Pacific Knowledge Acquisition Workshop (PKAW'04), in conjunction with The Eighth Pacific Rim International Conference on Artificial Intelligence, August 9-13, 2004, Auckland, New Zealand, 87-101.
[15] Collins, H.: “What is tacit knowledge”, in Schtzki, T. Knorr, C. & von Savigny, E. (Eds) The practice turn in contemporary theory, Routledge, London and New York, 2001.
[16] Grünbacher, P., Briggs, R.: “Surfacing Tacit Knowledge in Requirements Negotiation: Experiences using EasyWinWin”, Proc. 34th Hawaii International Conference on System Sciences, Hawaii, USA, 2001.
[17] Ryan, K.: “The Role of Natural Language in Requirements Engineering”, Proc. First IEEE International Symposium on Requirements Engineering (RE’03), San Diego, Ca. USA, 1993.