The Case for Dumb Requirements Engineering Tools

Daniel M. Berry[1], Ricardo Gacitua[2], Pete Sawyer[2,3], Sri Fatimah Tjong[4],

[1]Univ. of Waterloo, CA; [2]Lancaster Univ., UK; [3]INRIA Paris — Rocquencourt, FR; [4]Univ. of Nottingham Malaysia, MY

Abstract

Context and Motivation

This talk notes the advanced state of the natural language (NL) processing art and considers four broad categories of tools for processing NL requirements documents. These tools are used in a variety of scenarios. The strength of a tool for a NL processing task is measured by its recall and precision.

Question/Problem

In some scenarios, for some tasks, any tool with less than 100% recall is not helpful and the user may be better off doing the task entirely manually.

Principal Ideas/Results

The talk suggests that perhaps a dumb tool doing an identifiable part of such a task may be better than an intelligent tool trying but failing in unidentifiable ways to do the entire task.

Contribution

Perhaps a new direction is needed in research for RE tools.

Natural Language in RE

A large majority of requirements specifications (RSs) are written in natural language (NL).

Tools to Help with NL in RE

There has been much interest in developing tools to help analysts overcome the shortcomings of NL for producing precise, concise, and unambiguous RSs.

Many of these tools draw on research results in NL processing (NLP) and information retrieval (IR) (which we lump together under "NLP").

NLP-Based Tools and RE

NLP research has yielded excellent results, including search engines!

This talk argues that characteristics of RE and some of its tasks impose requirements on NLP-based tools for them and force us to question whether …

for any particular RE task, is an NLP-based tool appropriate for the task?

Categories of NL RE Tools

Most NL RE tools fall into one of 4 broad categories (a–d):

a. tools

- **to find defects and deviations from good practice in NL RSs, e.g., ARM and QuARS, and**
- **to detect ambiguous requirement statements, e.g., SREE and Chantree's nocuous ambiguity finder.**

Categories Cont'd

b. **tools to generate models from NL descriptions, e.g., Scenario and Dowser.**

c. **tools to discover trace links among NL requirements statements or between NL requirements statements and other artifacts, e.g., Poirot and RETRO.**

d. **tools to identify the key abstractions in NL pre-RS documents, e.g. AbstFinder and RAI.**

Key Needed Capability of Tools

Except for an occasional tool of category (a), part of whose task may include format and syntax checking …

each RE task supported by the tools requires *understanding* the contents of the analyzed documents.

Can Tools Deliver Capability?

However, understanding NL text is still way beyond computational capabilities.

Only a very limited form of semantic-level processing is possible [Ryan1993].

“I Know I’ve Been Fakin’ It”

Consequently, most NLP RE tools …

use mature techniques for identifying lexical or syntactic properties, and …

then *infer* semantic properties from these.

That is, they *fake* understanding.

Lexing in Category c

E.g., in a category (c) tracing tool, …

lexical similarity between two utterances in two artifacts leads to proposing links between the pairs of utterances and the pairs of artifacts.

Drawbacks of This Lexing

If the tool's human user (a requirements analyst) sees no domain relevance in the lexical similarity, then he or she rejects the proposal (imprecision).

Moreover, lexical similarity fails to find all relevant links (imperfect recall).

Recall and Precision

Recall **is the percentage of the right stuff that is found.**

Precision **is the percentage of the found stuff that is right.**

Validation and Interaction

Consequently, a human user always has to check and validate the results of any application of the tool,

and NL RE tools are nearly always designed for interactive use.

Using an Interactive Tool

In interactively using any tool, *e.g., a tracing tool*, that attempts to simulate understanding with lexical or syntactic properties, …

the user has to know that the output probably will

- **include some false positives (impresision) and**
- **not include some true positives (imperfect recall).**

Using an Interactive Tool, Cont'd

The action the user takes depends on

the cost of failing to have the correct output,

***i.e., the links that show the full impact of a proposed change*,**

vs. …

the costs of

- **finding the true positives and**
- **eliminating false positives**

manually.

In General, Though

Finding the true positives …

is usually both harder and more critical…

than eliminating false positives

for the tool's purpose.

(Hence the point size difference on the previous slide!)

Scenarios of Tool Use

Consider an analyst responsible for formulating a RS for a system (*S*).

The paper describes two scenarios:

1. ***S*** **does not have high-dependability (HD) requirements.**

2. ***S*** **has HD requirements.**

Scenarios of Tool Use, Cont'd

A system with HD requirements is one that is safety-, security-, or mission-critical.

We ignore Scenario 1 in this talk and focus on Scenario 2 (the more controversial and discussion provoking one ☺)

Second Scenario

The analyst is responsible for formulating a RS for *S with* HD requirements.

Second Scenario, Cont'd

In Scenario 2, …

A complete analysis of all documents about *S* is essential …

to find *all*

- **defects,**
- **abstractions,**
- **traces or modeling elements, and**
- **relationships**

that are present or implicit in the documents.

Normal Behavior of Analyst

Normally, the analyst would do the entire analysis manually.

The analyst has the uniquely human ability to

- **extract semantics from text and**
- **to cope with context, poor spelling, poor grammar, and implicit information (all too hard for NLP techniques).**

Analyst's Human Potential

Thus, with appropriate knowledge, training, and experience, …

the analyst has the potential to achieve

- **100% recall and**
- **100% precision.**

A Human is Human, Nu?

Of course,

- **a human suffers fatigue,**
- **and his or her attention wavers,**

resulting in

- **slips,**
- **lapses, and**
- **mistakes.**

In short, humans are fallible [DekhtyarEtAl]. Gasp!!!! … Oy, Gevalt!

Even worse!

The development of a HD *S* usually requires copious documentation, …

making fatigue and distraction so likely that …

tool support looks really inviting!

Second Scenario with Tools

Consider Scenario 2 vs. the 4 tool categories:

a. **tools to find defects and deviations from good practice in NL RSs,**
b. **tools to generate models from NL descriptions,**
c. **tools to discover trace links among NL requirements statements or between NL requirements statements and other artifacts, and**
d. **tools to identify the key abstractions from NL documents.**

Categories (a) & (b)

Tools in these categories can be useful despite the imprecision and imperfect recall.

See the paper.

Basically, we expect less than perfection from these tools; so we naturally work with and around them.

Category (a)

The paper shows how a tool of category (a) with less than 100% recall overall could have 100% recall on an identifiable subset of the defects, and thus could be useful in Scenario 2.

See the paper.

Category (b)

The paper shows how a tool of category (b), which is for sure less than perfect, is nevertheless useful for what it shows, simply because no one expects or requires it to be perfect.

See the paper.

Other Categories are Different

But, the quality of the output of tools of categories (c) and (d) have a direct effect on the quality of the system under development.

Category (c)

For a HD system, the tasks that depend on tracing are critical.

E.g., it is critical to find all of a security requirement's dependencies to ensure that a proposed change cannot introduce a security vulnerability.

To avoid manual tracing, 100% recall is required of a tracing tool.

Category (c), Cont'd

The fundamental limitations of NLP $\Rightarrow$ 100% recall is impossible, …

short of returning every possible link, …

which leads to complete manual tracing anyway.

Thus, automatic tracers are *not* well suited to HD systems.

Category (d)

The set of abstractions for a HD system are the bones of its universe of discourse.

For a HD system, the set of abstractions needs to be complete, to avoid overlooking anything that is relevant.

Category (d), Cont'd

Again, the fundamental limitations of NLP $\Rightarrow$ 100% recall is impossible, …

again, short of returning every possible abstraction, …

which again leads to complete manual finding.

Thus, automatic abstraction finders are *not* well suited to HD systems.

Verdict

Tools of categories (c) and (d) offer no advantage for HD systems, for which the *completeness* (as well as the correctness) of a tool's output is essential.

Naive Use Even Worse

As Ryan [1993] observed, naive use of such a tool may

1. **worsen the analyst's workload — the analyst looks at the tool's output and then has to do the whole manual analysis anyway**

   or

2. **lull the analyst with unjustified confidence in the tool's output.**

Rethinking Any NLP-Based RE Tool

If the tool cannot save the analyst work …

by doing 100% of analysis, and …

the analyst must manually analyze the whole document anyway, …

it might be best to forgo the tool and …

focus on doing the manual analysis very well.

Rethinking, Cont'd

Preparing to do well might include getting a good night's sleep the night before!

How to Use an Imperfect Tool

The second risk (lulling) of naive use of a tool with recall < 100% suggests that the best time to use such a tool is *after* a best-effort manual analysis that is felt to have been as thorough as possible.

After Manual Analysis is Done

Now, anything that the tool finds

1. **that the analyst overlooked or**

2. **that prompts the analyst to find something he or she overlooked**

is a low-cost bonus.

But …

But, if the user *knows* that a tool *will* be used later, then he or she may nevertheless fall into the trap of being lulled!

Another Source of Same Recommendation

This recommendation is consistent with Dekhtyar *et al.*'s observation that …

when asked to vet traces proposed by an automatic tracer, a category (c) tool, humans tended to decrease both the recall and precision of the traces.

Knowing that a tool was used made them sloppier.

Novices' Use of a Tool

Kiyavitskaya *et al.* have shown in an experiment that a high-precision, low-recall tool for annotating laws helps novices achieve 96% recall relative to experts.

I guess that the high precision helped the novices learn what is right, so that each could use his or her intelligence correctly.

Experts' Use of Same Tool

Experts did not participate inKiyavitskaya *et al.*'s experiment.

My bet is that …

Experts using the tool will find their recall deteriorating.

We need to test.

Another Idea

When no tool can do analysis *A* with 100% recall, …

but there is an algorithmically *identifiable* part of *A* that can be done with 100% recall by some tool *T*, then …

it might be useful to build *T* and let it do what it can, …

so that the analyst can focus on only the part of *A* that cannot be done with 100% recall.

The Key of the Idea

The key here is that the tool's and the human's parts of *A* are algorithmically identifiable, and …

the tool's and the human's parts of *A* together are *all* of *A*.

So that the analyst can *really* ignore the tool's part of *A*, and thus can *really* focus on the human's part of *A*.

SREE, An Example of Idea

Tjong's SREE, a category (a) ambiguity finding tool, finds …

only those potential ambiguities that are identifiable by a lexical scanner.

It leaves all other ambiguities to be found manually.

Use of SREE

SREE finds *all* potential instances of the ‘‘only’’ ambiguity by finding each sentence with the word ‘‘only’’.

The user quickly rejects false positives among these potential instances in a quick manual examination of the full list.

Use of SREE, Cont'd

Any ambiguity whose finding requires

- **parsing of NL sentences,**
- **correct part-of-speech identification,**
- **seeing context, or**
- **understanding semantics**

is left for manual searching.

SREE's Design Rationale

SREE has 100% recall for the ambiguities in its clearly specified domain, …

but less than 100% *precision* for these same ambiguities, …

since it finds, e.g., all instances of ‘‘only’’, not just the ambiguous ones.

SREE's Design, Con'd

The analyst can quickly eliminate the false positives in SREE's output

and then focus attention on the amgiguities that are outside SREE's clearly specified domain.

Enhancement of Dekhtyar & al

Humans vetting the poorer of two tools did a better job, as if they sensed the poor quality and rose to the occasion.

So maybe take the best tool available and randomly split its output to two groups of vetters.

BOBW!

Future Research Agenda

For each RE task to which NLP tools are being applied, e.g.,

- **abstraction identification,**
- **ambiguity identification, and**
- **tracing,**

Future Research Agenda, Cont'd

try to find an *algorithmically identifiable* partition of the task into

1. **a *clerical* part that can be done by a dumb tool with 100% recall and not too much imprecision and**

2. **a *thinking-required* part that must be left to a human analyst to do manually.**

Research Required

Finding this partition for any task will require research to think of a different way to decompose the task.

It will require a thorough understanding of the task and of what is algorithmically possible.

Research Required, Cont'd

For any task, the partitioning will take into account

- **the burden to the human analyst of the imprecision of the clerical part and**
- **the difficulty to the human analyst of the thinking-required part.**

Research Required, Cont'd

Obtaining this information will require research like that done by Dekhtyar *et al.* for tracing tools to determine

- **what is really difficult for humans and**
- **how well humans perform parts of the task with and without automation.**

Read Our Paper

Now go read our paper!

Write a rebuttal!

Join in on the research!

But, please be polite and stay for the rest of the talks of this session!