The University of Sydney

The Role of Computer Science in e-Science

Professor Jon Patrick

Chair of Language Technology

Sydney Language Technology Research Group

School of Information Technologies

University of Sydney

ScamSeek Project – Learnings from Research Objectives and Industrial Obligations

~

Use of Scamseek as an exemplar of e-Science to understand how it might operate

The Features of the Technology

- 3 systems built for 3 Internet data types: Web Pages, C2, C3
- Separates texts with subtle differences
- Finds text classes of very low frequency in a collection
- Finds very small texts of interest
- Uses the meaning intentions of the authors – not the word strings
- Developed on principled grounds of e-Science: linguistics, computational theory (ML) & software engineering
- Shown to work effectively for a wide range of financial scams

Task Definition

- Australian Security and Investment Commission (ASIC) -
- Financial Scams
  - Illegal Offerings
  - Unlicensed Advisors
  - Share Ramping

ASIC's Surveillance Context – Web Pages

- Internet Surveillance Challenge
- Surf Days
  - 30 people for a day, every quarter
  - 5,000 docs vetted
  - Reduced successively to 1500, 200, 50, 20

Project Requirements

- Build 3 language based classifiers
- Time 15 months - 2003-2004
- Budget $2.2m gross
- Personnel - variously 7-14
- Performance 50% correct {on scams}
- Audit and install an operational system

Framing an e-Science Solution

- Task Type - Document classification
- Capturing rich/deep semantics
- Data is unseen - structure unknown
- Exploiting current levels of research into these problems

E-Science Requirements

- How to convert an experimental system into an operational system
- How to get researchers (hackers) to perform in a consistent and constrained context
- How to ensure feedback is targetted and appropriate

E-Science Research Tasks

- Problems
  - Unbalanced classes
  - Extract semantic content

- Strategic Questions
- What Theory (linguistics model) to use - model of meaning – SFL
- What Pragmatic/Realisation (language) model to use – feature identification
- What transformation model to use – mapping features to attributes
- What statistical model to use – Machine Learning :SVM
- At what performance threshold would the semantic model overtake the classical Baseline (BOW) model

E-Science- What data to use?

- Final 4 Class Distribution
- Scams 5.90%
  - (orig. 1.7%- what is a scam document?)
- Scam-OA 3.08%
- Scam-Like 24.56%
- Non-Scam 66.46%

Audit Results – 21/10/2003

		Computed Class: Scam	Computed Class: Non-Scam	TOTAL
ASIC Class	Scam	18	26	**44**
	Non-Scam	6	1525	**1531**

Audit: Precision =.75, Recall=.41, F=.53

Train: Precision =.74, Recall=.35, F=.48

ScamSeek – Web Pages: Final Status

- System narrows search space to find 4scams/5 docs.
- This is a 100-fold productivity gain
- System has correctly identified many incorrectly classified docs in training corpus
- System found 4 missed scams in audit corpus
- System has semantically modelled & rendered as semantic networks in XML:
  - 20 types of scams
  - 4 SFL grammar networks: Expansion, Determiners, Modality, Polarity, (Processes)
  - (semi-)generic Persuasion
  - Weak modelling of 50 other registers

Results from 3 Corpora - 30th June 2004

	Web Pages	Unigrams 5000 atts.	Chan 2 (Unigram)	Chan 3 (Unigram)
Precision	74.4%	52.0%	85.0% (80.3)	85.2% (79.7)
Recall	52.8%	48.0%	83.4% (81.8)	63.9% (31.2)
F-value	61.8%	49.9%	84.4% (81.0)	73.0% (44.9)
Texts	373/6391		686/1483	1395/ 13716

Project Evaluation

- Performed beyond contract specifications
- Came in under time
- Came in under budget
- Saves the OZ public tens of millions of dollars per annum
- Received the Australian National Science Prize in 2005

Subsequent Comparative Study

	Prior Manual Operation	Scamseek Jan-Feb 2005
Labour Hours	190	Nil (=1900)
Documents retrieved	4,000	40,000
Of interest	285	257
Suspect	10	45

ScamSeek Organisation

- Client team
- Linguists team
- Computational Linguists team
- Software Engineering team

Software Engineers' Processes

- Design & create system experiment architecture – compute once principle
- Design and create system architecture
- Design and create linguists tools
- Install production system
- Generate Production system automatically

Automatic Production System Generation

- ALL processing modules and XML description models are stored in CVS
- Each experiment has a registry entry for each & every processing module, experiment model (parameters), SFL data model, and results
- Production System Generation is automatic and consists of collecting them all together
- A correct production system can be generated for every experiment ever executed

The Lessons for CS in E-Science

- Support a complete edifice of ICT for e-Science
  - Constructing workbenches for creating and managing permanent repositories that truly enable
    - sharing and review of research data and theoretical models
    - with high levels of access and interactivity
- The Results would increase
  - productivity and auditability of the science

An Architecture for Machine Learning Applications

- Current practice in the use of machine learning leads generally to weak science and even poorer reproducibility of results, because
- the number of transformations in the data create a distance between the theory under scrutiny and the testing thus weakening the generalisability of the results and even the meaningfulness of the testing.

The science derived from ML has poor reproducibility

- 1. a good deal of pre-processing goes into the preparation of the data prior to use in the computational learning stage that is not reported because it is (unjustifiably) considered unnecessary to reveal, that is, it is mere detail.
- 2. the choice of machine learner is varied depending on the researchers desire to find the "optimal" solution, thus ensuring that the statistical conditions of the testing is not properly factored into the scientific conclusions
- 3. the results between experiments and experimenters computed with different machine learners, even though they might be the same algorithm, are not necessarily comparable.

Standardisation of e-Science Software Environments

- Standardisation of reporting not methods is required. Not as it has been in the past, academic publication, but rather the provision of computational models executable on a standardised computer platform using the research data collected (not massaged) by the researcher.

- Implementation of such an environment requires a non-intuitive change in procedures for publishing scientific results.
- Researchers will have to produce industrial strength software that will execute their models in a standardised environment.
- It is unlikely that most researchers will have the software engineering skills to achieve this task so it must be provided by the national body for the preservation and dissemination of computationally modelled science.

A National Repository

- the researchers in a field need to contribute the software, in which their models execute, into open-source projects that will get them up to a professional software engineering standard and then loaded into a repository, along with the concomitant data, for dissemination and reuse.

Generic Model of Industrial Machine Learning - 1

- *Data-centric Users*
- *Model-centric Users*
- *Machine Learner Researcher-Engineers*
- *Software Engineers*

Generic Model of Industrial Machine Learning - 2

- *Data-centric Users* are those users whose interest is in the collection of the data and manipulation of the data and its consequences for explaining the natural or engineered world.
- *Model-centric Users* are those users who have a specific domain of expertise but who specialise in the computational modelling of that domain.
- These two User groups have the greatest interest in Active Learning

Generic Model of Industrial Machine Learning - 3

- *Machine Learner Researcher/Engineers* are responsible for the development of algorithmic statistics embedded in software that others use and for applying those models to suitable and appropriate tasks.

- *Software Engineers* are the experts who understand the technical issues and have the skills to make the modelling systems serve the needs of the external community, by designing them to be the most computationally efficient and by making them readily usable by the Data-centric Users.

Active Learning

- Support for continuous revision of data and models needs to be provided through ***active learning mechanisms***
- To achieve active learning as a matter of course for all machine learning and computational modelling is a software engineering task of significant intellectual demand.
- **However active learning is a key foundation of the enhancement of scientific endeavour through the systematic exploitation of ICT.**
- Active learning is performed by all user roles: service-centric, data-centric and domain-centric in the model revision process, and so it has to be designed into each of their processes. It is an s.e. intellectual challenge as to how to do what is ostensibly the same process for the different roles in the theory producing process

E- Science (AU) - **Proposal 1**

- The fundamental notion of an e-Science IT infrastructure is that a National Centre for Text and Data Mining be created. Its role would be to act as the lead organisation in creating:
- a programme of open source software development
  - for the care and sharing of scientific data and models in the national repository
  - for the research, development and adaptation of tools for the analysis of text and data.

E- Science (AU) - **Proposal 2**

- An organisational structure that recognises the presented model and centres the operational task for delivering ICT to e-Science as an exercise in appropriate advanced level software engineering of machine learning/computational modelling workbenches .

E- Science (AU) - **Proposal 3**

- The key work program of the National Centre of Text and Data Mining is the design and implementation of methods of active learning methods to be integrated with machine learning and other computational modelling where appropriate.

E- Science (AU) - **Proposal 4**

- The national repository can be constructed as a virtual system, extant on the computers of every participating organisation. The contents would be managed and curated by the local managers through the management software created by the National Centre for Text and Data Mining. Administrative strategies for getting scientists to lodge their content could be readily legislated for and/or be the conditions of scientific grants.

The Lessons for CS in E-Science

- A national repository for data and data models would allow for a leap in the degree of sharing of data and importantly data models.
- Further enhancement with a clear policy that the national repository develop and adapt machine learning software for general research analytics with an open source policy.
- Greater enhancement of this approach would be achieved by incorporating the functions of the national repository into the national grant allocations systems.
- Further advantage by creating an outlet for "grey science"

The End

- Gold standard repositories can be continually enhanced readily by active learning
- “Grey Science” can be contributed to a national repository
- National Science IT effort can be assessed and audited
- Contribution of data and models can be obligatory for public grants
- Recipient orgs given a brief hiatus to organise IP retention if necessary.
- Grid Computing is only a method for large scale computing - it is NOT doing e-Science
- Open Source Projects is the ONLY way to do this.