

**Machine
Learning for
Automated
Cross-
Referencing**

Sarah Bull

COMP3760

Background

Legal documents are very "human": they are a matter of great importance to humans, and need to be expressed in the utmost regularity possible, so that humans are able to read and understand them. Because of the precision needed, they are created and drafted by humans.

However, humans also display a *lack* of precision; for example, on the official government site for accessing legislation, ComLaw (www.comlaw.gov.au), the HTML is not very regular across different Acts, which were obviously created at different times by Parliament and placed into the system at these different times. It is worth contrasting to the Tasmanian government site (<http://www.thelaw.tas.gov.au/index.w3p>), where the legislation is clearly expressed in XML with useful cross-referencing, anchor tags, and 'point-in-time' searching available (<http://www.thelaw.tas.gov.au/about/enact.w3p>). (I have even found unclosed tags in the ComLaw legislation.) Proper structuring and cross-referencing of legislation has application to searching (Moens 2002) as well as improved access. As well as the Tasmanian project, there has been some Italian work done on structuring legislation into XML for easier access (Marchetti et al 2002).

There are three main types of legislative documents: official Acts (which are very structured), Regulations (also very structured), and Bills (which may be structured, as they are often drafts of Acts). (There is also Hansard material.) Besides legislative documents, other legal documents are cases and journal articles. Cases may be very structured; for example, looking at over 1300 patent decisions made by the Australian Patent Office between 1981 and 2008, I have found that there are less than 100 headings which occur regularly if similar keywords are combined (such as "law on extensions of time"/"law on extensions of time under regulation"). (Decisions sourced from Austli 2008.) Journal articles are also often structured using headings, and refer to cases and legislation. Using machine learning to properly structure these legal documents for cross-referencing can aid reading and searching for the human reader.

Humans may never be fully replaced regarding the drafting of and accessing legal documents, but machine learning can help in this task.

Problem/Task description

I plan to use machine learning methods such as genetic algorithms generating regular expressions and hidden Markov models to structure and cross-reference legal documents. These methods will be compared for efficacy and a software artefact using the results will be created. Legislation in particular is the focus of this project, with cases and journal articles also studied as time permits. Programming will involve Java and Python.

The legal documents in question are principally legislation, with extension to cases (especially patent cases), and journal articles. I aim to go 'further' than the Tasmanian legislation; for example, to locate definitions and penalty sections, and more strongly cross-reference amendments tables linking back to the amending Acts.

I have so far looked at using regular expressions to replicate the Tasmanian legislation's anchor tags. The automatic generation of regular expressions is an ongoing research problem; Li et al (2005) have used genetic algorithms to generate regular expressions for human gene splice sites, and Cetinkaya (2007) has written on generating regular expressions through grammatical evolution, with four out of five runs resulting in an optimum solution before 250,000 fitness evaluations. I aim to use similar genetic algorithm methods to generate regular expressions matching the required sequences in legislation structure.

Hidden Markov Models and CRFs are also key tools at NICTA SML. I plan to use software written at SML to test these algorithms for the task, and compare them to the evolved regular expression results for evaluation and research purposes.

The resulting artefact will use the best results out of the Tasmanian replication work, regular expression generation through genetic algorithms, and HMM/CRF results.

Risk Potentials:

- The genetic algorithm results are unsatisfactory
 - This is still a research opportunity
- The HMM/CRF results are unsatisfactory
 - This is still a research opportunity
- Both genetic algorithm and HMM/CRF results are unsatisfactory
 - This is still a research opportunity to compare the results and analyze the reasons for the failure
 - Software artefact will focus on legislation and build on the existing work done on ComLaw and Tasmanian results
- The software artefact is more difficult to construct than expected
 - The software artefact will focus on legislation and do less extension into cases and journal articles
- The computer will crash
 - There is a 99.99% chance that the computer will crash during the project because of various NICTA work being done, but (as experienced in previous crashes) backups are made on the NICTA servers. Notes and records are also stored on the Internet.
 - Results to long-running experiments will be incrementally stored in files in case of crashes occurring midway.

Plan/Schedule

8 August: Completed work of looking at Tasmanian legislation and replicating XML results

15 August: Project Plan due; Tasmanian results used on ComLaw files to get a hand-coded baseline and regular expressions

22 August: Getting some worthwhile results from regular expression generation [if results are poor, I will attempt to establish whether this was due to the unsuitability of the algorithm for this purpose or coding issues]

29 August: Continuing to work on regular expressions; applying regular expression generation to cases, eg patent cases, and more as time permits

5 Sept: Partial write-up of work so far, beginning the draft of the final report; begin work on learning about HMM/CRF methods

19 Sept: Have begun application of HMM/CRF methods

26 Sept: Continuing application of HMM/CRF methods

3 Oct: Getting some worthwhile results from HMM/CRF methods [if results are poor, attempt to establish whether this was due to the unsuitability of the algorithm for this purpose or coding issues]

10 Oct: Developing the artefact; doing further work on the algorithms used and extending them to other legal documents as time permits

24 Oct: Artefact and final presentation/report to be in a state consistent with established deadline [if computer crashes, backups are stored on NICTA Linux Sydney servers and on Google servers]

31 Oct: Final presentation due this week

7 Nov: Final report due this week

References

Australasian Legal Information Institute, 2008, Joint Faculty of UTS/UNSW Faculties of Law, Australian Patent Office decisions at <http://www.austlii.edu.au/au/cases/cth/APO/>

Cetinkaya, Ahmet, Jul 2007, "Regular expression generation through grammatical evolution", *GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, ACM

Li, Jing-Jing, Huang, De-Shuang, MacCallum, RM, and Wu, Xiao-Run, 2005, "Characterizing human gene splice sites using evolved regular expressions", *IJCNN '05, IEEE International Joint Conference on Neural Networks*, Vol 1, pp 493-498

Marchetti, Andre, Megale, Fabrizio, Seta, Enrico, and Vitali, Fabio, Apr 2002, "Using XML as a means to access legislative documents: Italian and foreign experiences", *ACM SIGAPP Applied Computing Review*, Vol 10 Issue 1

Moens, Marie-Francine, Jun 2005, "Combining structured and unstructured information in a retrieval model for accessing legislation", *ICAIL '05: Proceedings of the 10th International Conference on Artificial Intelligence and Law*