A Right to Access Implies A Right to Know: An Open Online Platform for Research on the Readability of Law

Michael Curtotti*
Eric McCreath[o]

** Legal Counsel, ANU Students Association & ANU Postgraduate and Research Students Association, PhD Student, Research School of Computer Science, Australian National University*
[o] *Lecturer, Research School of Computer Science, Australian National University*

Abstract. The widespread availability of legal materials online has opened the law to a new and greatly expanded readership. These new readers need the law to be readable by them when they encounter it. However, the available empirical research supports a conclusion that legislation is difficult to read if not incomprehensible to most citizens. We review approaches that have been used to measure the readability of text including readability metrics, cloze testing and application of machine learning. We report the creation and testing of an open online platform for readability research. This platform is made available to researchers interested in undertaking research on the readability of legal materials. To demonstrate the capabilities of the platform, we report its initial application to a corpus of legislation. Linguistic characteristics are extracted using the platform and then used as input features for machine learning using the Weka package. Wide differences are found between sentences in a corpus of legislation and those in a corpus of graded reading material or in the Brown corpus (a balanced corpus of English written genres). Readability metrics are found to be of little value in classifying sentences by grade reading level (noting that such metrics were not designed to be used with isolated sentences).

Keywords: readability, legislation, legal informatics, corpus linguistics, machine learning, natural language processing, readability metrics, cloze testing

1. Background and Motivation

We are embedded in a network of legal rules. We are not always able to understand those rules. Sometimes social heuristics or specific training

(as, for example, in road rules) enable us to understand and comply with law. Often considerable expense is invested in 'explaining' the law to citizens: such as through official government information supplementing legislation, or through investment of private resources in legal services. As citizens we often need to know, and are entitled to know, the law which affects us. In a democratic context, legal rules are theoretically the outcome of consultative processes in which the entire community has a voice and in which the interests and views of the members that make it up are given due recognition and protection.

The internet has transformed the way in which society engages with legislation. It has changed how legal professionals access the law. As significantly, it has expanded and changed the audience which accesses and reads legislation. The Declaration on Free Access to Law states that public legal information is digital common property and the common heritage of mankind and calls for law to be accessible to all on a non-profit basis and free of charge.[1] This Declaration is made in the context of the considerable effort by LII's and others to achieve the practical realisation of such free access.(Martin; J., 2005)

In the UK, the Office of Parliamentary Counsel is pursuing a 'Good Law' initiative, a key objective of which is to make law more usable. The UK First Parliamentary Counsel observed:

> *Legislation affects us all. And increasingly, legislation is being searched for, read and used by a broad range of people. It is no longer confined to professional libraries; websites like legislation.gov.uk have made it accessible to everyone. So the digital age has made it easier for people to find the law of the land; but once they have found it, they may be baffled. The law is regarded by its users as intricate and intimidating*.(OPC-UK, 2013)

They note that while in the past readers of UK legislation tended to be legally qualified, that is no longer true. They report an audience of two million unique visitors per month for the legislation.gov.uk site.(OPC-UK, 2013) Similarly in the NZ case the users of legislation has broadened: *It*

[1] http://www.worldlii.org/worldlii/declaration/.

seems once to have been supposed that law was the preserve of lawyers and judges, and that legislation was drafted with them as the primary audience. It is now much better understood that acts of Parliament (and regulations too) are consulted and used by a large number of people who are not lawyers and have no legal training. There the government legislation website received 30,000 unique visitors per month.(NZ, 2008, p 14)

In 2008, the New Zealand Law Commission and the New Zealand Parliamentary Counsel's Office together undertook an inquiry into the Presentation of Law starting from the proposition that: *'It is a fundamental precept of any legal system that the law must be accessible to the public.'* Their inquiry identified three aspects of access to law: availability to the public (such as hard copy or electronic access), 'navigability' - the ability to know of and reach the relevant legal principle, and finally accessibility in the sense of the law '*once found, being understandable to the user.'* (NZ, 2008) The issues paper which preceded their report put it more succinctly:

> *Citizens should be able to know and understand the law that affects them. It is unfair to require them to obey it otherwise. This is an aspect of the rule of law.*(NZ, 2007)[2]

Concepts of 'understandability', or this third category of accessibility, are closely related to the concept of readability which is the subject of this paper. DuBay reviews a number of the definitions that are offered for readability: 'readability is what makes some texts easier to understand than others'; 'the ease of understanding or comprehension due to the style of

[2] Interestingly is difficult to find this principle clearly enunciated in primary sources (for example in human rights documents). An example that approaches it may be found in article 14.3 of the International Covenant of Civil and Political Rights which provides the right to be informed of charges in a language the individual understands, and the right to a free interpreter). The New Zealand Commission and Parliamentary Counsel note that in their case there is no principle of statute law that 'it must be understandable'. (NZ, 2008) Nonetheless 'understandability' is a guideline is to Departmental officers and drafters involved in the creation of legislation: "*For legislation to command public acceptance it must meet certain standards. It must be developed in accordance with proper processes, reflect legal principle, be technically effective, and be able to be understood by those to whom it applies. NZ Legislative Advisory Council Guidelines on Process and Content of Legislation".*

writing'; 'ease of reading words and sentences' as an element of clarity; 'the degree to which a given class of people find certain reading matter compelling and comprehensible'; and 'The sum total (including all the interactions) of all those elements within a given piece of printed material that affect the success a group of readers have with it. The success is the extent to which they understand it, read it at an optimal speed, and find it interesting'.(DuBay, 2004) There is some variance in these definitions but they have in common (explicitly or implicitly) orientation to the needs and characteristics of a given group of readers and they assume that it is possible for a writer, by changing the selection and organisation of words, to communicate essentially the same concepts while facilitating understanding.

Kohl carries out a study of the principles of accessibility in the context of online publication of foreign laws. She notes the existence of two rationales for accessibility (including in the sense of an ability to 'know' the law). Firstly, it is unfair for a citizen to be subject to liabilities if they are unable to know the law. This rationale focuses on human and societal values. Secondly, the purpose of the law maker is to achieve compliance with law, and thus the law maker wishes it to be known. From this viewpoint, the regulator's interest in administrative effectiveness and efficiency is a motivation for ensuring access and knowledge. She notes that although legal jurists and courts propound the principle that laws should be clear or understandable as an element of the rule of law, a failure of clarity does not necessarily result in relief from legal detriment: it may amount to a *moral* principle but its effect in law is uncertain. (Kohl, 2005)

Milbrandt and Reinhardt argue for the existence of a right to access the law (in the broader sense of physical or electronic access). Principles of the rule of law, freedom of information, and principles of human rights such as the right to freedom of expression and to an effective remedy imply rights to access and know the law. Like others, they explore scenarios where access is effectively denied.(Milbrandt and Reinhardt, 2012)

A stream of action to improve the readability of law is associated with the plain language movement that particularly gathered steam during the early 1990s. Proponents of plain language cite extensive empirical studies validating the benefits of plain language for the understanding of

text. This extends to the legal context, including through widespread support of plain language measures adopted by legislative drafting offices.(Kimble, 1994) As one legislative drafting office puts it in their plain language manual:

> *We also have a very important duty to do what we can to make laws easy to understand. If laws are hard to understand, they lead to administrative and legal costs, contempt of the law and criticism of our Office. Users of our laws are becoming increasingly impatient with their complexity. Further, if we put unnecessary difficulties in the way of our readers, we do them a gross discourtesy. Finally, it's hard to take pride in our work if many people can't understand it.*(OPC-Australia, 2003)

The influence of the plain language movement has seen it mandated in both legislation and executive orders: "A number of federal laws require plain language such as the Truth in Lending Act, the Civil Rights Act of 1964, and the Electronic Funds Transfer Act. In June 1998, President Clinton directed all federal agencies to issue all documents and regulations in plain language."(DuBay, 2004)

Above we have seen both principle and practice directed to making the law more accessible in the sense of its ease of comprehension. Yet, despite this an observation made three decades ago by Bennion, the author of a leading text on statute law, could just as appropriately be made today:

> *It is strange that free societies should thus arrive at a situation where their members are governed from cradle to grave by texts they cannot comprehend*.(Bennion, 1983, p 8)

Existing empirical research on the readability of legislation supports a conclusion that legislation is inaccessible to large proportions of the population - that for many citizens it is very difficult or incomprehensible. This research moreover suggests that even plain language does not significantly alter this reality. (See discussion below in Section 3.)

The various rationales for accessibility in the sense of 'understandable' text, as discussed above, coupled with the limited progress towards its effective realization, motivates the work reported in this paper. The work is concerned, particularly from a computational perspective, with

identifying appropriate measures and approaches for assessing the readability of legislation and implementing computationally based tools for carrying out readability research on legislation. In section 2 we describe both well established and newer approaches for assessing readability including traditional readability metrics, human-centred evaluation and natural language processing and machine learning. Section 3 reviews existing research on the readability of legislation. These two sections provide a baseline for further research that might be undertaken on readability of legislation.

Section 4 describes the development of an online platform for readability research, which is offered as an open service for researchers interested in carrying out readability research. The development of this platform is part of a broader body of research on the development of computational tools for reading and writing law.[3] The platform is made available to any researchers who may wish to carry out readability research on legislative materials (or indeed any other text). The plat- form provides a number of readability tools. A tool is provided for the extraction of readability metrics from text. A second tool is designed to enable "cloze testing" (a method widely agreed to be an accurate method for measuring the readability of text). The site also provides a tool for carrying out subjective user evaluation of a text. Finally, the platform provides access to natural language processing facilities which can be used for extraction of a variety of language features such as parts of speech and ngrams.[4] The tools are accessed through a straight forward interface and are accompanied by documentation to facilitate usability.

In section 5 we report the application of this platform for initial investigations on three corpora: a corpus of graded readers, the Brown Corpus and a corpus of Australian federal legislation.

Leaving aside the theoretical justifications that might be advanced to support this view, the axiomatic position taken by this paper is that all

[3] For details see http://cs.anu.edu.au/people/Michael.Curtotti.
[4] An ngram is simply a sequence of a given length e.g. a bigram is a sequence of two letter, two words, or two parts of speech.

individuals subject to law are entitled to know its content and therefore to have it written in a way which is reasonably accessible to them.

2. Approaches to Assessing Readability

In seeking to enhance the readability of legislation, a question which naturally arises is how to assess whether given text is 'readable' or 'more readable'. Within a computational context we are particularly interested in the potential for enhancing the assessment of readability through application of computational techniques. Readability metrics naturally suggest themselves as an area of investigation, given their widespread use.

While readability metrics, such as the Flesch metric are well known (for example incorporated into Microsoft Word), their reliability and relevance are disputed both within and beyond the legislative context. Apart from such metrics, a number of other possibilities exist: user evaluation (such as comprehension testing or cloze testing and more recently crowdsourcing) and application of techniques arising from recent natural language processing and machine learning studies of readability.

2.1. Readability Metrics

Reading measures such as the Flesch, Flesch-Kincaid, Gunning, Dale-Chall, Coleman-Liau and Gary-Leary are among the more than 200 formulas which have been developed to measure the readability of text. These formulas (although varying in formulation) address two underlying predictors of reading difficulty: semantic content (i.e. the vocabulary) and syntactic structure. Vocabulary frequency lists and sentence length studies both made early contributions to the developments of formulas. The Flesch formula calculates a score using average sentence length and average number of syllables per word as measures for determining text difficulty. Formulas of this kind are justified on the basis of their correlation with reading test results. For example, the Flesch formula correlated at levels of 0.7 and 0.64 in different studies carried out in 1925 and 1950 with user tested texts.(DuBay, 2004)

The uses and abuses of such formulas have been widely debated. An important observation in this context is that these tests were not conceived as measures of comprehensibility of text, rather they were designed to help teachers select appropriate texts for children of different ages.(Woods et al., 1998)

In 1993 an Australian Parliamentary Committee report on clearer legislation (having reviewed use of readability metrics) commented:

> *Testing for the readability of legislation by using a computer program is of limited value. The most effective way of testing legislation is to ask people whether they can understand it - a comprehension test. Ideally this type of testing should occur before the legislation is made.* (Melham, 1993)

Evidence presented to the Inquiry included the view that research had undermined the validity of readability metrics and the view that readability metrics could mislead by mis-categorising the complexity of legislative sentences (Melham, 1993, p. 98).

A review of methods for measuring the quality of legislation carried out in New Zealand observed that readability metrics can only play a limited screening role in the prediction of readability. It considered such metrics to have limitations such as not detecting how complex ideas are, whether the language is appropriate to the audience or whether a sentence is ambiguous. They note that legislative drafters in the UK have concluded that such tests do not measure readability in a comprehensive sense, but that they seem reasonably good as an initial indicator of problematic text.(PCO-NZ, 2011)

Despite their limitations, readability metrics are used in practice and have a body of supporting research. They have been influential and continue to be widely used:

> *Writers like Rudolf Flesch, George Klare, Edgar Dale, and Jeanne Chall brought the formulas and the research supporting them to the marketplace. The formulas were widely used in journalism, research, health care, law, insurance, and industry. The U.S. military developed its own set of formulas for technical-training materials. By the 1980s,*

> *there were 200 formulas and over a thousand studies published on the readability formulas attesting to their strong theoretical and statistical validity* (DuBay, 2004).

A debate carried out between a readability specialist, computer scientists and others in the context of computer documentation is illuminating as to the limitations of readability metrics. Klare, the readability specialist participating in the debate, cited a number of limitations of readability metrics. These included that they function best as screening devices only, need to be interpreted in light of reader characteristics, cannot be used as formulas for writing style 'since changes in their index variables do not produce corresponding changes in reader comprehension' and should be used in conjunction with other approaches such as use of human judges, cloze procedure and usability testing. Further, readability metrics are designed for larger blocks of text providing a connected discourse and won't work well on disconnected fragments or single sentences (something relevant to the experiments reported below).(Klare, 2000) Others note the poor correlation between different readability metrics themselves.(Woods et al., 1998) Beyond this, some studies have found poor correlation between human judgements as to readability and the scores assigned by readability metrics(De Clercq et al., 2013; Harrison and McLaren, 1999; Heydari and Riazi, 2012). Heydari et al. observation perhaps sums up the state of research:

> *If any conclusion is possible to draw from the hodge-podge of studies done on readability formulas, it is that there are two opposite views toward the use of them. Both of these two views have been advocated by different researchers and there is enough empirical evidence for each to be true. Thus, it can be declared openly that the formulas have both advantages and disadvantages.* (Heydari and Riazi, 2012)

With such conclusions, some caution is required in using readability metrics. The caution is reinforced in respect of legal language, particularly legislative language. Little validation has been undertaken of readability metrics in the context of legal language. Until that validation is carried out and the parameters of valid application understood, any conclusions based on application of such metrics must be qualified with uncertainty. Their advantage is that they are readily calculated without significant investment of human resources - a factor that has likely

contributed to their widespread use. The Readability Research Platform includes tools for extracting various readability metrics.

2.2. COMPREHENSION TESTING, CLOZE TESTS AND CROWDSOURCING

In this section we review some human centred approaches to evaluating the readability of text. Such methods equate to the field of user evaluation, in human computer interaction. Such methods are perhaps the most promising for application to improving the readability of legal language. If properly implemented, such tests can measure how understandable text is to readers, and can be targeted to particular reader groups of interest (e.g. the general public or individuals particularly affected by an item of legislation). Their disadvantage is that they are resource intensive to carry out, while crowdsourcing requires access to platforms with large user traffic and programming skills.

2.2.1. *Comprehension Testing and User Evaluation*

A traditional method of testing the ability of a reader to understand a text is to administer a comprehension test. This method can be used in reverse to assess the difficulty of the text, for given populations of readers. Tests are deployed by having a student read a passage and then answer multiple choice questions regarding its content.(DuBay, 2004)

2.2.2. *Cloze Tests*

The cloze procedure involves testing the ability of readers to correctly reinsert words that have been deleted from a given text. Typically the test is administered by deleting every *n*th word in the text. When used to assess the readability of a text the cloze procedure is administered by deleting every fifth word (including sometimes five different versions of the text staggering the deletion), and replacing it with a blank space, which the reader must fill in by guessing the missing term (Bormuth, 1967). Although initially conceived as a remedy for the shortcomings of readability formulas, the cloze procedure came to complement conventional reading tests (DuBay, 2004). Cloze procedure was also developed to provide a more valid measure of comprehension than traditional multiple choice comprehension tests.(Wagner, 1986) Of greatest interest in this context is use of cloze tests as a measure of the readability of a text. Bormuth notes that there is a high correlation

between cloze readability testing and comprehension testing on human subjects:

> *A reasonably substantial amount of research has accumulated showing that cloze readability test difficulties correspond closely to the difficulties of passages measured by other methods.* (Bormuth, 1967)

Bormuth cites studies, including his own, which show correlations ranged from .91 to .96 with the difficulty of texts assessed with traditional comprehension tests.(Bormuth, 1967) When properly applied the cloze test provides an indicator of how difficult a text was for given readers. A cloze score of below 35% indicates reader frustration, between 35% and 49% is 'instructional' (the reader requires assistance to comprehend the material) and 50% or above indicates independent reader comprehension.(Wagner, 1986)

As we see below (section 3), the cloze procedure has been used as a means of assessing the readability of legislation. The Readability Research Platform described below includes a cloze tool, which is in demonstration phase.

2.2.3. *Crowdsourcing*

The emergence of large populations of online users, opens the possibility of such users being engaged in the task of assessing the readability of legislation. A parallel might be drawn with crowdsourcing used to support scientific research such as through the Zooniverse platform, some projects of which use human judgements to support the classification of images of galaxies, to cite one example.[5] De Clercq et al. undertake an evaluation of the effectiveness of crowdsourcing as a method of assessing readability. They compared the accuracy of crowdsourced human judgements of the readability of text with those of expert judges, finding a high level of agreement in readability ranking between the experts and crowdsourced users. crowdsourced users were presented with two randomly selected texts of one to two hundred words and invited to rank them by readability. Expert teachers, writers and linguists were given a more complex task of assigning a readability score to each presented text. In addition to concluding that crowdsourced user judgements and expert judgements were highly correlated as to readability ranking, they found

[5] How Do Galaxies Form Classification Project https://www.zooniverse.org/project/hubble.

that readability metrics had a lower correlation with these two judgement sets.(De Clercq et al., 2013)

A more general study by Munro et al. on the use of crowdsourcing in linguistic studies concluded that there was a high correlation between traditional laboratory experiments and crowdsourced based studies of the same linguistic phenomena. Among their conclusions was that crowdsourced judgements closely correlated with cloze testing results, which as we have seen above is a key approach to undertaking readability studies. (Munro et al., 2010) We are unaware of any studies which have used crowdsourcing to assess the readability of legislative text. There does not seem to be any serious impediment to using such an approach and the Readability Research Platform includes a demonstration tool for collecting user evaluations of text.

2.3. Machine Learning and Natural Language Processing

Recent years have seen a growing body of research seeking to apply natural language processing and machine learning to assessing the readability of text. The term 'natural language processing' represents the capacity of computers to hold and analyse large bodies of text. Natural language processing can be applied to represent text as collections of characters, collections of words, to annotate words with their grammatical type (such as noun, verb, adjective etc.), to aggregate words into grammatical phrases and to represent the syntax of sentence as a grammatical tree. Such purely functional annotation can be extended to information extraction - the identification of entities such as persons, organisations, places etc, and the identification of relationships. Such work falls under the heading of natural language processing.

Machine learning is grounded in mathematical theory and provides well elaborated processes of enabling patterns to be learnt from a given body of data. Data (for example linguistic data) is represented as a set of 'feature', 'value' pairs associated with each item from the dataset. For example a sentence has associated with it a set of features such

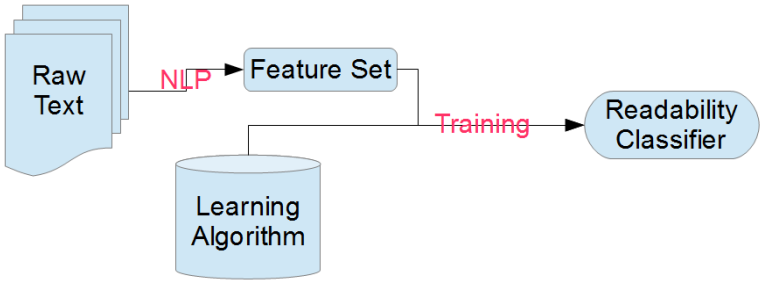

Fig. 1. A typical natural language processing and machine learning pipeline in application to readability

as its length, the number of words, the parts of speech of those words, the given vocabulary and patterns such as the occurrence of two words in sequence. Such features can then be used to learn a model which with a known level of accuracy predicts (for example) the classification of a previously unseen sentence. Machine learning includes both 'supervised' and 'unsupervised' learning. In supervised learning a data set already labelled with the appropriate classifications is provided as input to the learning algorithm. In the unsupervised case the machine learning is carried out on unlabelled data.[6]

Readability research has applied both these processes to seek to automatically predict the readability of given text. A pipeline of transformations are carried out on a dataset consisting of input documents (which need be no longer than a single sentence) with the aim of learning a capacity to predict the readability of given text. Figure 1 illustrates a typical process, the desired end result of which would be a learned classification model with the capacity to correctly classify text for its readability with a known level of accuracy.

Many have in common the hypothesis that 'deeper' language features provide valuable data for the task of assessing the readability of text.

---

[6] See Bird et al. for a very accessible and practical introduction to natural language processing. Chapter six also introduces machine learning in application to the classification of text.

An exhaustive review of the application of these techniques to readability is not carried out here but a number of aspects of particular interest are highlighted. A key question is what features might assist us in assessing readability? Studies have systematically examined sets of features for their utility in assessing readability. The most straight forward features examined have been readability metrics themselves and 'surface' features such as average sentence length, average word length and average syllable length, capitalisation, punctuation. Other features studied include lexical features such as vocabulary and type/token ratio, [7] parts of speech frequencies, ratio of content words to function words, distribution of verbs according to mood, syntactic features such as parse tree depths, frequency of subordinate clauses, ngram language models, discourse features, named entity occurrences, semantic relationships between entities and anaphora occurrences. (Dell'Orletta et al., 2011; Kate et al., 2010; Feng et al., 2010; Si and Callan, 2001)

Collins Thompson and Callan in 2004 undertook a study of the use of 'language models' to predict reading grade. They build a model of grade language based on the probability of a word for each grade level. This approach was based on the observation that the probability of a word occurring in a text varies depending on the grade level of the text. However the authors were guarded in the conclusions they felt able to draw as to the effectiveness of their approach (Collins-Thompson and Callan, 2004).

Schwarm and Ostendorf in 2005, also used a language modelling approach, in combination with other features. They apply a support vector machine algorithm to undertake machine learning using features such as readability metrics, surface features, closeness of match for language models built on graded reading material, parse tree heights and number of subordinating conjunction. Their support vector machine grade prediction outperformed the Flesch-Kincaid grade measure and the Lexile measure by a wide margin. None of the features they used stood

[7] A 'type' is say the word 'red' and a token is any word. So in the phrase "the cat sat on the mat" the type to token ratio is 5/6, as the word 'the' occurs twice.

out as critical to classification, but removal of any degraded performance.(Schwarm and Ostendorf, 2005)

Heilman et al. in 2008 test a number of machine learning algorithms using unigram language models and full and sub-tree features as grammatical input. They attain an accuracy of 82% in predicting grade level of documents in their corpus using a combination of language features.(Heilman et al., 2008)

Pitler and Nenkova also in 2008 use adult reading materials from the Wall Street Journal graded as to readability by human judges. They note that 'readability' assessments are dependent on audience and note that graded readers designed for language learners are not generalisable to the question of general readability of more standard texts. They assess various features for predicting readability using this labelled corpus. Surface, syntactic, lexical cohesion, entity grids and discourse relations. They identify discourse relations as most predictive of readability (correlation of .48), followed by average number of verb phrases, followed by article length. Combining the various features they examined attained the highest accuracy of around 88%. Surface features (which underlie most readability metrics) they find to be poor predictors of readability.(Pitler and Nenkova, 2008)

Feng et al. undertake a study of similar scope to Schwarm noted above. Again using a corpus of graded material they seek to identify factors most predictive of readability. They find parts of speech features (particularly nouns) to be highly correlated with grade level. They also note that among surface features used in traditional readability metrics, average sentence length has the highest predictive power.(Feng et al., 2010)

Kate et al., like the Pitler study, use a labelled dataset of adult reading materials. The dataset of 540 documents is labelled by expert and naive human judges. The machine learning algorithm is then trained to predict readability from a training set labelled with expert judgements. The authors find that using diverse linguistic features, they are able to exceed the accuracy of naive human judges as to readability. As with other studies combining features produced the highest levels of accuracy.(Kate et al., 2010)

Aluisio et al. also apply machine learning and like other studies find that combining linguistic features increases accuracy of prediction. They are also concerned to leverage readability assessments for the task of simplifying text. (Aluisio et al., 2010)

Of particular interest for classifying the readability of legal rules are readability studies which focus on classification of single sentences or shorter text fragments. As legal rules are often written as single sentences may be of greater assistance than readability measures which focus on paragraphs or blocks of text. Dell'Orletta et al. carry out readability assessment at both document and sentence level, undertaking a binary 'hard' vs. 'easy' classification of Italian texts. As with other studies they examine a wide range of features. However they also are particularly interested in assessing features that might later be applied to the process of text simplification. Base features (such as underlie readability metrics) show little discriminative power for sentences, but they find that the addition of morpho-syntactic and syntactic features increases accuracy of sentence level classification to 78%.(Dell'Orletta et al., 2011; Sjoholm, 2012)

Sjoholm's 2012 thesis also addresses predicting readability at sentence level. He notes the absence of existing metrics for predicting readability at sentence level. He builds on previous studies by developing a probabilistic soft classification approach that rather than classifying a sentence as 'hard' or 'easy' gives a probability measure of membership of either class.(Sjoholm, 2012)

The application of natural language processing and machine learning to the task of predicting readability has made considerable progress over the last decade or so. Studies such as those above have demonstrated that prediction of readability can be significantly improved by incorporating higher level linguistic features into predictive models. Further, of interest to us, the Dell'Orletta and Sjoholm studies underline the inadequacy of traditional readability metrics (as they are based on surface features) for assessing readability at sentence level. It is also notable that only initial steps have been taken to apply findings in this field to identifying reliable methods of improving readability.

Natural language processing and machine learning, as suggested by the progress of recent research, offers considerable promise that it may allow progress in understanding and addressing readability issues in legislation. Significant is still required to adapt the existing research to application to readability in the legislative field. A limitation of such methods is that without a considerable body of labelled data, it is difficult to attain high levels of accuracy with machine learning. Obtaining reliably labelled data is best achieved through user studies of the kind described in Section 2.2. Another challenge inherent in machine learning is determining those 'features' which are most associated with readability. The work reported above provides some guidance as to which features may prove useful.

3. Empirical Research on the Readability of Legislation

In section 1 we noted the extensive attention given to readability of legislation by government agencies and the plain language movement. Readability is a standing concern of legislative drafting offices with plain language being a frequent goal or commitment of such offices. (Kimble, 1994; OPC-Australia, 2003) Here we seek to summarise the findings of empirical research which directly assesses the readability of legislation. Such empirical studies are limited in number and scope, though considerable work has been undertaken on tax legislation.

An early example was a study reported in 1984 in which cloze testing was undertaken on several samples of legal text including legislative language. 100 generally highly educated non-lawyers (28% had undertaken some postgraduate training) were tested. The group averaged 39% accuracy, a result close to 'frustational' level for cloze testing. Ten participants who had only high school education experienced greater difficulty, averaging 15% – a result consistent with total incomprehension.(Benson, 1984)

In 1999, Harrison and McLaren studied the readability of consumer legislation in New Zealand, undertaking user evaluations, including the application of cloze tests. They seek to answer a number of questions including: how comprehensible to consumers and retail workers is New Zealand's consumer legislation? The study found traditional readability metrics to be unreliable. The results of cloze testing on extracts from the legislation led to the conclusion that the legislation would require

explanation before being comprehended at adult level. For young adults (aged 18-34), comprehension levels were even lower (within the frustrational level). Paraphrase testing, where participants were asked to paraphrase the legislation, also showed that participants found the Act difficult to understand with one section proving almost impossible to access. Participants complained of the length of sentences and most felt there was a need for some legal knowledge to understand the text. All felt the text should be made easier. The researchers also inferred from cloze testing that simpler terms were required in the legislation to make it more accessible to the public.(Harrison and McLaren, 1999)

In the early 1990's Australia, New Zealand and the United Kingdom pursued tax law simplification initiatives which involved rewriting at least substantial portions of tax legislation. The goal in Australia's case was stated to be to 'improve the understanding of the law, its expression and readability'. Cloze testing on a subset of the work was however inconclusive, finding participants found both the original language and the rewritten language difficult.(James and Wallschutzky, 1997) Smith et al., reviewing the effectiveness of the same program, concluded that results fell 'far short of an acceptable bench-mark'. They used the Flesch Readability Score as a measure of readability finding that readability of sections of tax law replaced in the tax law improvement program, improved on average from 38.44 to 46.42 - a modest improvement. The result is well short of the general Flesch benchmark of 60-70 for readability. i.e. even after improvement, the legislation remained difficult to read. Over 60% of the revised legislation remained inaccessible to Australians without a university education.(Smith and Richardson, 1999) A similar study of the readability of goods and services tax legislation in Australia also applying the Flesch Readability Index, finds an average readability of 40.3 (i.e. low). Again such results exclude considerable proportions of the Australian community.(Richardson and Smith, 2002)

A study in Canada carried out usability testing on plain language and original versions of the Employment Insurance Act. Members of the general public and expert users were recruited to carry out testing. All participants completed more questions in the plain language version. Similarly all participants using the plain language versions were more accurate in their answers. All respondents, particularly those from the general public, found navigation and comprehension difficult irrespective

of version. They also found that for all versions respondents faced difficulty in understanding the material. These findings indicated that in this instance while plain language reduced difficulty it did not eliminate it. Nonetheless participants preferred the plain language version and found it easier to use.(GLPi and Smolenka, 2000)

Tanner carried out empirical examination of samples of Victorian legislation, assessing them in light of plain language recommendations of the Victorian Law Reform Commission made 17 years earlier. The authors noted that the Law Reform Commission had recommended that *on average* sentences should be no longer than 25 words and that complex sentence structure was to be avoided. In a study of six statutes they found that the average sentence length was almost double that recommended by the Commission, and that over time sentence length had increased. In the Fair Trading Act (a piece of legislation of general importance to citizens), they found that the number of sentences with six or more clauses was particularly high. Although they also note improvement in some areas, they conclude: "The net result is that many of the provisions are likely to be inaccessible to those who should be able to understand them. This is because the provisions 'twist on, phrase within clause within clause'."(Tanner, 2002)

An empirical study of the usability of employment legislation in South Africa also found that respondent accuracy improved considerably with a plain language version of the legislation. The respondents who were drawn from year 11 school students averaged a score of 65.6% when tested on the plain language version, whereas the control group scored an average of 37.7%. Like other studies it found that plain language improved comprehension.(Abrahams, 2003)

A 2003 review of the Capital Allowances Act in the UK which was rewritten as part of the UK's tax law improvement program undertook interviews with a number of professional users. These professionals in general responded that the new legislation was easier to use and more understandable.(OLR, 2003)

A similar review of the Income Tax (Earnings and Pensions) Act also carried out in the UK again found that the interviewed group (primarily tax professionals), were largely positive about the benefits of the

simplification rewrite, expressing the view that the revised legislation was easier to use and understand, although also noting the additional costs of relearning the legislation.(Pettigrew et al., 2006)

A 2010 study of the effects of the tax law simplification in New Zealand employed cloze testing to determine the degree to which the simplification attained its goals. They cite a 2007 Australian study by Woellner et al. which using cloze procedure, found that novice users of both original and amended versions did not achieve benchmark comprehension but found the new legislation (ITAA 1997) marginally easier (35% vs 24%). In their own study they reported that most of their respondents (mainly respondents unfamiliar with the tax system) found the cloze testing either difficult or extremely difficult. They found that the older (unamended) Act was the least difficult - a finding contrary to their expectation given prior research in New Zealand - this they attributed to the nature of the selections from the older legislation. The overall average cloze results was 34.17, with unfamiliar respondents achieving 30.86%. They note that less than 25% of their subjects were able to exceed the instructional guideline of 44%. (Sawyer, 2010)

The empirical readability research points to two conclusions. Firstly writing in plain language assists comprehension of legislation. Secondly legislation is generally incomprehensible or difficult to read to large sections of the population, even in those cases where plain language revision has been undertaken.

4. An Open Online Platform for Readability Research

4.1. Motivation and Description of the Platform

The previous sections of this paper provides an overview of the body of knowledge which provides context for the Readability Research Platform, which is maintained on an Australian National University server accessible via the internet[8] and which is described below. Its particular purpose is to enable an extension of the reported research on readability of legislation (and other texts for that matter), initially to meet the needs of

[8] http://buttle.anu.edu.au/readability/.

the authors, but later as an effort to make relevant tools available to other researchers. In this context, a number of factors contribute to the design of the tool:

- The primary use case for which the platform is designed is carrying out readability research (including on legislation).

- Given this, the platform needs to facilitate or enable the application of various readability approaches. It thus includes tools that cover the various approaches discussed above. It is also extensible, as additional tools can readily be added as need arises. The availability of these tools in one place facilitates comparative studies of different approaches, as well, it is hoped, as facilitating comparison of work undertaken by different researchers using the tool.

- The community interested in the readability of law is a multidisciplinary one. In this context the platform would preferably be accessible to researchers with little or no experience of programming. For this reason the protocols adopted in the platform are as simple as possible, avoiding frameworks that require familiarity with particular representations of data. The tool accepts plain text as its primary form of input and seeks to simplify the steps required to extract data.

- Given the scale of legislative data, the platform be capable of handling either large documents or a large number of smaller documents at a practical speed.

- The platform would ideally enable researchers to build on existing research, making it important to incorporate access to natural language processing tools, which are at the cutting edge of readability research.

- The design of the tool should enable collaboration with interested researchers through potential for integration with online legislative sites.

- The tool would ideally facilitate the reproduction of existing results in the readability field.

Apart from its use for research, the demonstration pages on the website provide visual introductions to the readability tools they demonstrate.

Where available, the platform makes use of existing open access libraries for carrying out underlying natural language processing, while abstracting away details of use of these packages in application to readability tasks. Natural language processing is provided by either the NLTK Language Toolkit or Montylingua.(Bird et al., 2009; Liu, 2004) Most readability metrics are extracted using a plug in to NLTK developed by Thomas Jakobsen and Thomas Skardal. http://code.google.com/p/nltk/source/browse/trunk/nltk_contrib/nltk_contrib/readability/

Fig. 2. The Readability Research Platform Website

4.2. Using the Readability Research Tool

The site provides a number of demonstration pages illustrating the kinds of outputs that can be extracted using the platform (see Figure 2). These include: readability metrics, natural language processing, cloze testing and user evaluation. A help page is provided which is designed to address the needs of researchers. The page describe commands that can be sent to the server which returns either data extracted from text provided as input or html (that can be used as a widget in another web page). These tools are intended primarily for the purpose of data extraction from text. Data

that can be obtained includes readability metrics, surface features, parts of speech, chunk phrases and ngram data. The data is returned as text which can either be saved to file or used as input to code developed by the researcher.

The server will respond to a http request sent to the server in for- mats described on the help page. Also the server functionality can be explored manually using the browser's url address box. For example typing: http://buttle.anu.edu.au/readability/?getariXXXXThe brown fox is quick.', and sending it to the server, will return the ARI readability metric for the sentence: 'The brown fox is quick.' A list of available commands and their descriptions is provided at the website help page.

The primary scenario for which to the platform is designed is automated extraction of data from text. While it is possible for a researcher to cut and paste text into the tool, this is impractical in most real world research scenarios. In order to retrieve data the researcher can use simple scripts which send http requests to the server and retrieve the requested data. The retrieval of data can be achieved in a few lines of code. The key steps in a typical use case scenario are:

1. create a local file into which to save results;

2. send a command (any arguments) and the text to be analysed to the server;

3. save the response from the server to the local file;

4. analyze resulting data using an external statistical package.

Two examples of simple scripts written in Python are provided in Appendix A which illustrates these steps. If the resulting data is comma delimited and saved into a file with a .csv extension, it can be opened in Microsoft excel and analysed or subjected to further processing.

A more complex example of use of the Readability Research Platform is provided in Appendix B. The consists of the calls made in the iPython command line interface, a script and a class for saving data into the Weka Machine Learning Software data format 'ARFF'. The example in Appendix B, which is written in Python, can be replaced with code

written in another programming language. The resulting datafile could then be used for carrying out machine learning using Weka package.

4.3. TESTING AND PROFILING

Unit testing was carried out on individual metrics to ensure the code behaves as intended. The Selenium testing platform was used for these tests, which confirmed the accuracy of a number of readability metric results on short input texts.

Also performance profiling was completed on a variety of the natural language related commands to understand and compare their performance characteristics. This was done by providing the server with a document and timing how long the server took to complete the test for a variety of different configurations. The documents had word counts ranging from 100 to 1000 in increments of 100. The results are graphed and shown in Figures 3 and 4.

The graph in Figure 3, using a logarithmic scale, shows the large range in performance for different processing tasks. Extraction of British National Corpus Metrics (which was slowest) took in the order of 10s of seconds, whereas the simple ARI metric takes tenths of a second to process on similar sized documents.

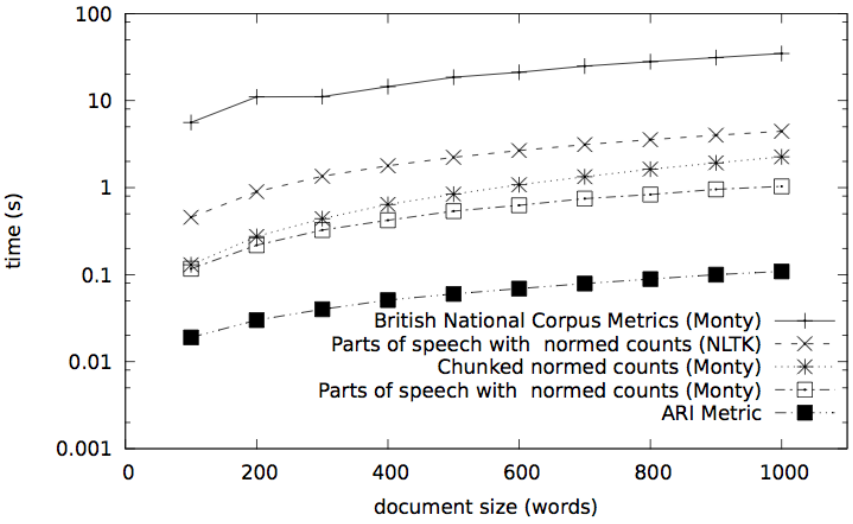

Fig. 3. Log Time Performance of Selected Data Extraction Commands by Document Size

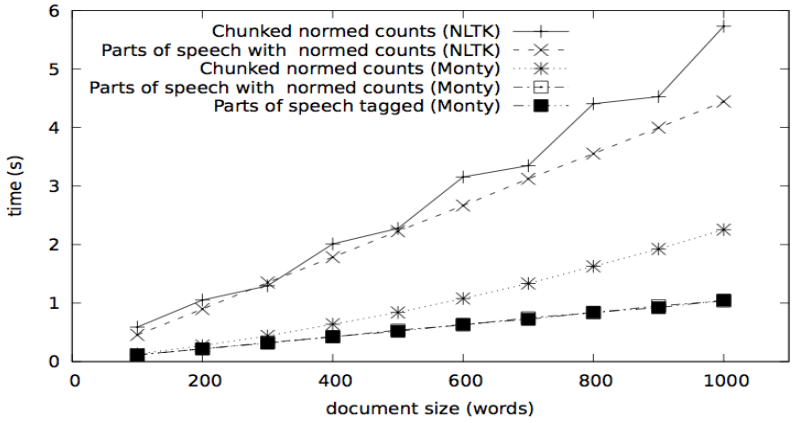

Fig. 4. Scaling of Performance by Document Size

The graph in Figure 4 shows that the parts of speech processing are linear with respect to performance. This would suggest these evaluations would be viable for large documents. Note that the Montylingua tool performed better than NLTK for the processing parts of speech by a factor of approximately 4.3. Also from this graph it is clear that the chunking code contains some quadratic scaling, this indicates the evaluation may be problematic if the documents become very large. There was little difference in performance between raw or normed counts so we have only graphed the normed count versions.

The speed of the platform, although far from instantaneous, is sufficient for a wide range of realistic research scenarios. For example extracting parts of speech counts for a 1,000,000 word corpus using the NLTK option (one of the slower commands) would take about an hour and a quarter. A significant factor in performance is the inherent computational complexity of tasks such as parts of speech tagging which are likely to already be optimized in the underlying code. Nonetheless, we have undertaken little work to optimize performance, a task that could be pursued as the platform is further developed.

5. Initial Investigations of Legislation and Readability using Machine Learning

The Readability Research Platform described above was used, through its http request protocols, to undertake initial investigations to characterise legislation for readability purposes. The focus of investigation was at the level of individual sentence or individual legal rule (the latter often constituting a single sentence in drafting practice). This enables us to investigate legislative language from the point of view of the citizen or user seeking to understand an individual rule or sentence.

We investigated a number of questions.

1. Do traditional readability metrics or surface features of a sentence assist us in assessing the readability of the sentence?

2. Does parts of speech or chunk data from a sentence assist in assessing its readability?

3. Do features such as the above provide us with a measure of whether legislative 'sentences' are 'normal' English?

Three corpora of English language were used to investigate these questions.

- A corpus of extracts from graded readers which was downloaded from the internet (graded reader corpus).[9]

- The Brown University Standard Corpus of Present-day American English which is a balanced corpus of English genres.(Francis and Kucera, 1964) The corpus is available through the Natural Language Toolkit.(Bird et al., 2009)

- A corpus of 'popular' legislation, identified as such on the official Australian legislation website (www.comlaw.gov.au), which was downloaded from that site and from the AustLII website (austlii.edu.au) and compiled into a corpus of legislation. Head material and appendices and notes were removed from the legislative corpus as such material does not form part of the legal rules themselves.[10]

5.1. Do Readability Metrics and Surface Features Assist in Assessing the Readability of a Sentence?

The Readability Research Platform [11] was used to extract readability metrics and "surface features" from individual sentences from the graded reader corpus. The resulting data file was in 'ARFF' format, and was used to carry out machine learning using the Weka Data Mining Software Package.(Hall et al., 2009) 'Classification' was used to explore how useful

[9] http://www.lextutor.ca/graded/. A copy of the graded corpus used in this research can be obtained at http://cs.anu.edu.au/people/Michael.Curtotti/data/gradedcorpus.zip.
[10] http://cs.anu.edu.au/people/Michael.Curtotti/data/legislativecorpus.zip.
[11] http://buttle.anu.edu.au/readability/.

the extracted features (in this case readability metrics and surface features) were for classifying the material into their correct grades.

Readability metrics are typically designed for use on passages of text of 100 words or more (as we discussed above). Even though they are not designed for the task of assessing readability of individual sentences, are they nonetheless useful?

The potentially limited value of such metrics for readability assessments at sentence level is illustrated by Figure 5, which was generated by the Weka machine learning package on data extracted from the Graded Reader Corpus. Each colour represents a distinct grade level, showing the distribution of Coleman Liau Index results for sentences for that grade. The extensive overlap of the metric's results for the different grades will be evident. The implication is that if all that is known about a sentence is its Coleman Liau Index, it will be very difficult to say which grade it comes from. Although the mean for the Coleman Liau distribution can be seen to move higher as the grade level increases, each grade level has a very similar range. This overlapping distribution is typical of what we observed with respect other readability metrics.

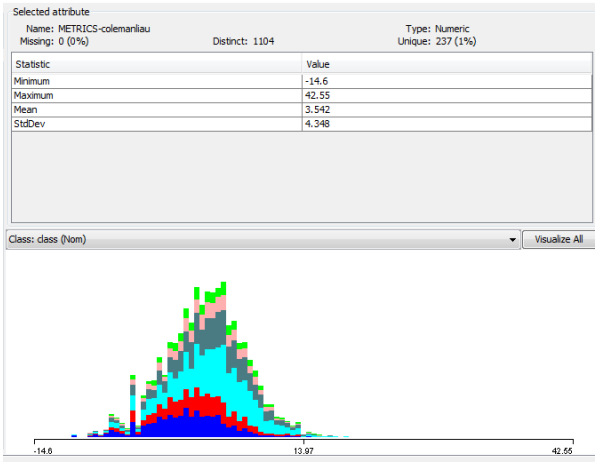

Fig. 5. Stacked Histogram Distribution Visualization of Coleman Liau Metric for Six Grade Levels from Graded Reading Corpus

We carried out multiclass classification on 14456 data items trialling a number of learning algorithms. The baseline accuracy value of 22.2% (ZeroR – i.e. guessing the most frequent class) was increased to 28.4% accuracy in the case of the Weka package support vector machine implementation (SMO) tested using ten-fold cross validation. The highest accuracy was 36% on any classification for any particular grade. By themselves, readability metrics are insufficient for the task of distinguishing reading grade level, at sentence level. Such metrics are not completely useless at sentence level either, however, as accuracy over the base level was increased by 6.2%.

5.2. Does parts of speech or chunk data from a sentence assist in assessing its readability?

Language may also be analysed by parts of speech (POS) (such as determiners, nouns, verbs, prepositions), and by phrase chunks (noun phrases, verb phrases, adjectival phrases and prepositional phrases).

The language features provided by POS and chunks, is additional to that provided by readability metrics. Do such features enhance classification of sentences by grade level?

We found that machine learning using these features alone, or these features in combination with readability metrics and surface features, does enhance the classification of sentences according to grade reading level.

Tests were carried out on a smaller set of 1613 data points drawn from the graded reader corpus with additional features and then machine learning classification was carried out using ten fold cross validation.

The baseline ZeroR accuracy was 19.9%. Machine learning using just parts of speech and chunk information increased accuracy to a maximum of 30.4%, using Bayesnet learning. Using parts of speech, chunking information and readability metrics and surface features as well as ranking and frequency information from the British National Corpus, increased accuracy to a maximum of 35.2%, using the Decision Table algorithm. Again ten fold cross validation was used for machine learning. In no case was accuracy on any particular grade higher than an F-measure

of 0.44. Accuracy increased by 15.3% over the base- line. Again we see that even with the additional features, classification results remain poor.

A qualifier with this particular trial is the significantly smaller number of data points used for the machine learning.

5.3. Do Readability Metrics allow us to reach conclusions as to whether legislative 'sentences' are 'normal' English?

Above we saw that readability metrics and surface features provide limited capacity to determine if a sentence belongs to a particular grade level. By contrast the same is not true of the ability to distinguish sentences drawn from legislation from other English sentences.

Legislative sentences, as characterised by readability metrics and surface features, are quite distinct from the graded reader material as illustrated by a visualization of a number of these metrics. In Figure 6 for each metric, legislative sentences (the top row in tan) are an outlier. The figure show the Weka summary visualization of the distribution of values for some of these metrics and the 'words per sentence' surface feature. From visual inspection it can be seen that the distribution of these metrics for each of the graded readers is similarly distributed, whereas legislative sentences have a much broader range of values.

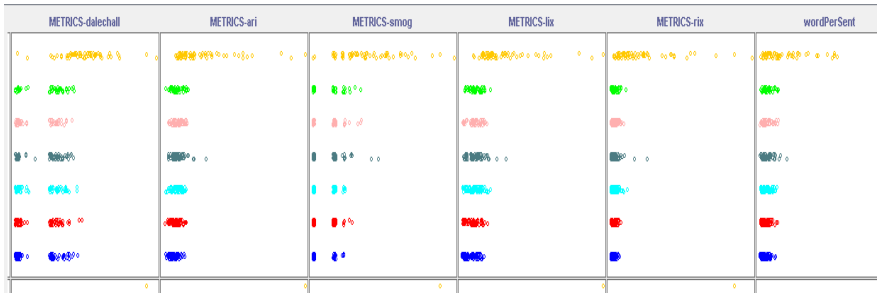

Fig. 6. Distributions of Metrics for Graded Reading Material and Legislation. The top row shows range of values for legislation for illustrated metrics, lower lines illustrate relative distribution ranges for graded readers.

The hypothesis suggested by this visualization is that legislation is significantly different from normal English usage. We may further hypothesise that this difference may contribute to reading difficulty for readers expecting to find 'normal English. Such a hypothesis would be consistent with the findings of studies that we have examined above that legislative texts are often inaccessible to non-professional readers.

The hypothesis suggested by the visualization is further supported by machine learning which we carried out on both the legislative corpus and the graded readers. Machine learning is far more effective at distinguishing legislative sentences from the graded readers. A balanced and randomized dataset was prepared which included both legislative sentences and sentences from the graded reader material. The dataset contained a total of 16 566 items. The ZeroR default accuracy was 17.9%. On this dataset machine learning algorithms increased accuracy to 30.7% (JRip), 34.4% (REPTree), 34.5% BayesNet, 34.9% (SMO), 34.1% (Decision Table) and 33.1% Naive Bayes. As with the Brown corpus comparison discussed below, the F-measure accuracy of classification of legislation was considerably higher than for readability grades: 0.87, 0.89, 0.79, 0.83, 0.83 and .80 respectively for the different learning algorithms. 0.37 was the highest F-measure accuracy for the classification of any grade level on any of the learning algorithms used.

A potential objection to the validity of this comparison is that the graded readers are not in themselves 'normal' or real world English. Especially at lower grade levels, the readers are simplified English produced for the purpose of assisting readers to develop their reading skills. A comparison is required with real world English.

To address this objection we also carried out a further comparison using the Brown Corpus which is a balanced corpus of different genres of English text: i.e. it is a representative sampling of the major forms of written English. Given that the Brown corpus is not organised by assumed difficulty of reading, we would expect that readability metrics would not be particularly useful in distinguishing different genres (not being designed for this task).

Again visualization (Figure 7) suggests that legislative sentences are an outlier. There is in this case more variance between the Brown Genres,

nonetheless legislative sentences have a much wider range of variation for readability metrics and surface features as compared to the genres.

The test carried out on the corpus confirmed this with JRip machine learning using readability metrics and surface features only increasing the base ZeroR figure from 9% to 10%. This result also allows a conclusion that the kinds of features that readability metrics provide are unable to distinguish between genres of English at a sentence level.

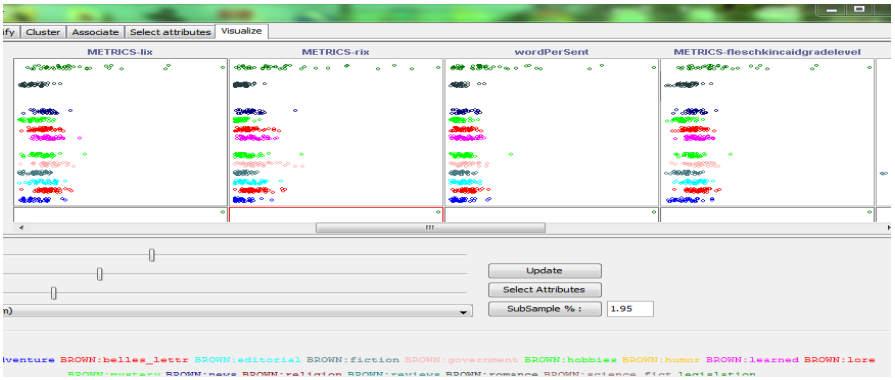

Fig. 7. Distributions of Metrics for Brown Genre and Legislation (the top row is Legislation). As with Figure 6 lower rows show relate metric value distribution, but in this case for Brown genes.

Testing with legislative sentences versus Brown genres are not as marked as the results with graded reading material, but nonetheless legislative sentences are the most distinctive genre by a large margin if compared with the genres in the Brown corpus. Whereas the F- measure for classifying Brown corpus genres does not rise above 0.17, for legislation the figure rises to 0.47, with a precision of 73% and a recall of 35%. The comparison with a balanced corpus of written English increases confidence that legislative language is indeed 'different' as far as readability metrics and surface features are measures of that difference.

Initial work was also undertaken to examine whether other features (parts of speech and chunk data), also suggest a significant difference in legislative language. A further set of experiments was undertaken

analysing a smaller dataset of Brown genres and legislation consisting of 3691 datapoints. JRip in this instance produced unreliable results as it dealt with legislation as a residual category into which otherwise unclassified items were labelled.

A number of different learning algorithms were therefore applied. Apart from JRip (and Conjunctive Decision Table, which also produced low results (11% overall accuracy)) each machine learning algorithm found it considerably easier to correctly classify legislative sentences as opposed to sentences from Brown genre categories, using parts of speech and chunk phrase data. (See Table I)

Machine Learning Algorithm	F-Measure Accuracy Legislative Sentences	Nearest or Highest result for Brown Genres	Overall Accuracy of ML Algorithm
ZeroR	0.13	0.00	6.93%
JRip	0.14	0.24	11.38%
NNGE	0.70	0.33	23.95%
Decision Table	0.69	0.28	22.30%
REP Tree	0.79	0.30	24.00%
J48 Tree	0.83	0.35	24.84%
SMO	0.85	0.44	30.94%
Nave Bayes	0.83	0.30	23.98%
BayesNet	0.85	0.41	28.34%
Lazy KStar	0.80	0.36	22.05%

Table I. Machine Learning Algorithm Accuracy Legislation And Brown Genres

Further indicators that legislation is different from the Brown genres in respect of its parts of speech and chunk characteristics came from a larger dataset extracted from the Brown Corpus and the Legislative Corpus. This dataset consisted of 31482 datapoints of which the legislative data constituted 3185 datapoints and the remainder from Brown genres. Using Weka, all features except parts of speech and chunk data were removed. Features not having discriminative power were also removed, leaving 43 features. Principal components analysis was utilised to represent features as independent orthogonal variables, leaving 36

features. Machine learning was carried out on this dataset with similar results as above.

Visualization of some of these principal components (see Figure 8), suggest that legislation can also be very different in its parts of speech and chunk characteristics to other English 'genres'. This complements the finding above that legislative readability metric and surface feature characteristics are different to 'normal' English. Further work is required to characterise the nature of these differences in detail and how they may be related to readability of legislation. They are suggestive that to the extent that 'plain English' has been achieved in legislation, (if it has) it has not resulted in 'normal English'.

The study we report above, has a number of limitations that fu- ture research might address. Only one jurisdiction is examined. The linguistic features examined are limited to readability metrics, surface characteristics, parts of speech and chunking data. The machine learning studies reported above show that other linguistic factors can be effective discriminators and also need to be explored in the legislative context

Fig. 8. Weka Visualizations of two principal components derived from parts of speech and chunk information (from left to right) for Brown Corpus Genres, Legislation Corpus and combined data

Every person who has read legislation knows that it is 'different'. What results such as the above show, is that it is possible to measure this difference. It is interesting that despite a commitment (and the considerable effort and expense in some cases) towards 'plain English' in

the drafting of laws, laws remain 'different' as a body of language (if we assume that the Australian Commonwealth legislative corpus is reasonably representative of legislative language in general). We are unaware of any past characterization of the empirical difference between a corpus of general English and a legislative corpus. An ability to define such points of difference, at a minimum can be envisaged to assist in identifying legislative sentences which are outside the umbrella of 'normal English usage'.

6. Conclusions and Future Work

This paper provides a background and context for carrying out readability research in application particularly to legislation with a particular focus on potential application of computational techniques. Empirical research on the readability of legislation supports a conclusion that most readers find it incomprehensible or difficult to read. Research on readability using natural language processing and machine learning is in its infancy, and is a promising area for further investigation. As far as we are aware there have not been significant studies on the readability of legislation applying crowdsourcing or machine learning techniques[12].

We report the development of the Readability Research Platform which is made available as an online service to researchers wishing to carry out readability research - whether on legislation (or other legal texts). We describe its envisaged use in a research context and report its performance characteristics.

Use of the Platform as a research tool is demonstrated in carrying out what is, as far as we are aware, novel empirical research assessing the difference between legislation and other written English using natural language processing and machine learning and examining readability metrics, surface features, parts of speech and chunk characteristics. Among our findings are that legislative data drawn from popular national

[12] Comparative corpora studies of legislation and other genres have previously been carried out in Dutch and Italian although not specifically in the context of readability issues.(van Noortwijk et al., 1995; Venturi, 2008).

legislation in one English speaking jurisdiction is different to 'normal' written English in respect of such characteristics at sentence level. Finding a difference is consistent with the empirical research which finds that legislative English is hard. How far we have come in achieving accessible legal language remains a live question. In addition, we undertake preliminary work on the use of parts of speech, chunk information, readability metrics and surface features to distinguish readability of sentences, using as input data, a corpus of graded reading material. This work shows such features to have discriminative value, but accuracy is low on a multiclass classification task. Readability metrics are, as others have observed, unreliable measures of readability, the more so in the context of legislation, given its difference from other English genres.

Finally, the establishment of the Readability Research Platform, we hope creates the potential (in combination with legislative sites and collaboration with other research groups) to carry out cloze testing and user evaluations on a large number of legal rules found in legislation. Such future studies, in our view, would be potentially make a valuable contribution to properly characterizing the readability of legislation. In particular, if a large dataset is created of legislative provisions labelled with reliable readability assessments, it can be expected to make available the full power of machine learning to identify those elements of legislative language which present a barrier to readability. At a minimum, it is likely to help us determine, with a greater level of confidence, how readable a particular piece of legislative text may be to its end users, without needing to undertake further human evaluations.

7. Appendix

These appendices provide examples of code used to run commands provided by the Readability Research Platform. Examples in Appendix A illustrate use of http requests to extract data. Appendix B provides python code to send multiple simultaneous commands and build a dataset for later machine learning.

A. Simple http examples

A.1. Single command with single input

This section illustrates sending a single command to the server using the iPython command line interface to send a command using python code. The output appears in blue. Line [1] imports the requests module which handles http requests. Line [2] defines the text to be analysed. Line [3] specifies which command is to be sent. Line [4] defines the url which is to be used (as described in the help page at the Readability Research Platform. Line [5] sends a http get request and saves the content to the variable 'output'. Line [6] prints the variable output to the screen. Lines [2]-[4] can be simplified to a single line but are expanded here to clarify the process.

```
Python 2.7.3 |Anaconda 1.4.0 (64-bit)

In [1]: import requests

In [2]: text = "The quick brown fox jumped over the lazy dog."

In [3]: command = "getallmetrics"

In [4]: url = 'http://buttle.anu.edu.au/readability/' + '?' +
                                          command + "XXXX" + text

In [5]: output = requests.get(url).content

In [6]: print request

fleschreadingease,fleschkincaidgradelevel,rix,colemanliau,

gunningfog,dalechall,ari,smog,lix::

103.70,1.03,0.00,4.43,3.60,0.45,6.62,3.00,9.00
```

A.2. Simple example using text file and input and saving results to output file for later processing

The example below illustrates a simple use case where data analysis is carried out on an input text file. The results are saved to a file that can be opened in excel.

```
# load python modules used in script
import requests
# open the text file to be used in read mode
textfile = open('demoparas.txt', 'r')
# split the document into a list of paragraphs
paragraphs = textfile.readlines()
# close the textfile - its not needed anymore
textfile.close()
# open a new datafile using .csv extension in write mode
# csv means a comma delimited file and can be read by excel
datafile = open('demoresults.csv','w')
# create an url & command variable
# ('?getari' and 'getfleshkincaidgradelevel' in this example)
url = ''http://buttle.anu.edu.au/readability/
commandurl1 = url + ''?getariXXXX"
commandurl2 = url + ''?getfleschkincaidgradelevelXXXX"
# loop through each paragraph and submit to
# the Readability Research Platform
# server, saving results to datafile
'∴n' inserts a line break after each data item
for para in paragraphs:
# get the results from each command
```

```
    result1 = requests.get(commandurl1 + para).content
    result2 = requests.get(commandurl2 + para).content

    # create a line to be written to the datafile

    results = result1 + ',' + result2 + '∴n'

    # print out to screen as well

    print results

    datafile.writelines(results)


# close the datafile

datafile.close()
```

B. Example Script and Code for Data Extraction from the Readability Research Platform (http://buttle.anu.edu.au/readability/)

B.1. Commands sent using iPython to run extraction script and the wekaTool, which saves data in Weka compliant format

The example below assumes that you have installed iPython, which makes running python code easier and comes with key libraries such as the Natural Language Toolkit already included. The text below is an illustration of the commandline interface in iPython with the two commands that would be needed to run the scripts and code in Appendix B.

```
Python 2.7.3 |Anaconda 1.4.0 (64-bit)

IPython 0.13.1 -- An enhanced Interactive Python.

[1] cd "D://YourDirectoryHoldingTheScripts/"
```

```
[2] run yourExtractionScript.py
```

B.2. EXAMPLE EXTRACTION SCRIPT

The following is an example of a script run to extract data by sending multiple commands to the Readability Tool. The script is run from iPython as illustrated in Appendix B.1. Copy and save the script with an appropriate name - 'yourExtractionScript.py'. In the following code, comments describing the code are in dark green and are not executed by the computer.

```
# load code for holding/processing data as Weka format

import wekatool as weka

import os, nltk

# The list of data commands to be sent to the server

commandList = [['getallmetrics'],['getsurfaceD','normed']]

commands = str(commandList)

#output file where results will be saved

outputfile = 'legislation1.arff'

# Load the wekaTool for later use

wkT = weka.wekaTool()

# Change to directory of your legislation corpus

os.chdir('D://PhD/A-Local/yourLegislationCorpus/')

# get the names of text files to be processed

filelist = []

for file in os.listdir("."):

    if file.endswith(".txt"):
```

```
            filelist.append(file)
    # for each text file process the file
    for file in filelist:
        # provide feedback on progress
        print "STARTING ON FILE: ", file
        #assign a class to data as required
        classType = 'legislation'
        f = open(file).read()
        # splitting the file into sentences
        sentences = nltk.sent_tokenize(f)
        count = 1
        #For each sentence in the file process the sentence
        for sentence in sentences:
            print "PROCESSING SENTENCE: ", count
            count +=1
            # run the weka tool to load
            # the data item for later processing
            wkT.loadTextData(sentence,commands,classType)
    # process the data and write it to file
    #  for  later  use  for  machine  learning
    arff = wkT.writeARFFfile(outputfile)
```

B.3. Example python code for extracting data in Weka Format

The following code can be used with the commands illustrated in Appendix B.1 and the script example in Appendix B.2. The entire code below can be saved into a file called 'wekaTool.py', after which can be called by code illustrated above.

```
from __future__ import division

import requests, urllib2, math, re, traceback, sys, ast

"""

A class for extraction of features from text.

This code is developed as part of PhD studies in the
ANU Research School of Computer Science.

It may be freely used for research purposes only.
For other uses, contact the author.

Author: Michael Curtotti 2013

"""

class wekaTool:

    """

    command syntax:

    [[command,mode,engine,type,ngramcount],[...],[...],...]

    the first value is required

    the 2nd to 4th values are optional

    mode = raw or normed

    engine = monty or nltk
```

```
type = letter or word or pos

test command = ['getposd','normed'],

['getchunkd','ra
w','monty'],

['getsurfaceD','ra
w','monty'],

['getngram','raw','monty','le
tter','1,2'],

['getngram','raw','monty','
pos','1,2,3']

# holds data for a single input after which it is cleared

featureDictionary = {}

# a holder for keys for features across many data items

featureList = []

# a holder for data extracted from text input

# holds multiple inputs for later data formatting

# inputs for each text item will be held as python

# dictionary objects with each key representing

# a feature and each value the value of that feature.

dataset = []

url = ""

errorCount = 0 inputCount = 0

def                                          __init__(self,
url='http://buttle.anu.edu.au/readability/'):
```

```
        """
        A class for creating a feature set from text. Supply
        the url for code testing purposes only
        """
        self.url = url
        self.errorCount = 0 self.inputCount = 0
    def loadFile(self, text = "", commands = [],
classType = "UNK"):
        loads an entire file, partitioning the input into
        sentences
        Used as alternative to the loadTextData function
        Needs ['partition'] to be included in
        list of commands
        """
        try:
            commands = str(commands)
            body =
            {'commands':[commands],'text':[text],'class':[cl
            assType]}
            result = requests.post(self.url,body).content
            assert not result.startswith('ERROR')
            processedresult = ast.literal_eval(result)
        self.dataset += processedresult
        self.inputCount += 1
    except Exception, e:
```

```
            self.errorCount += 1

            print "ERROR with input"

            print "Number of Errors: ", self.errorCount

            print    "Number    of    Successful    inputs:    ",
            self.inputCount print "TEXT WAS: ", text[:200]

            print "COMMANDS WERE: ", commands

            print traceback.print_exc()

            _,_,tb = sys.exc_info()

            traceback.print_tb(tb)

            print "====================="

    def loadTextData(self,text = "", commands = [], classType =
    "UNK"):

        """

        processes text data by calling the Readability Tool

        at   http://buttle.anu.edu.au/readability/   receives   data
        extracted from the input text and holds it

        for later output to file or
        printing

        """

        try:

            commands = str(commands)

            body                                                  =
            {'commands':[commands],'text':[text],'class':[classTy
            pe]}

            result = requests.post(self.url,body).content
```

```
            assert not result.startswith('ERROR')
            assert len(body.keys())>0
            processedresult = ast.literal_eval(result)
            if not len(processedresult.keys())==0:
                self.dataset.append(processedresult)
            self.inputCount += 1
        except Exception, e:
            self.errorCount += 1
            print "ERROR with input"
            print "Number of Errors: ",
            self.errorCount
            print "Number of Successful inputs: ",
            self.inputCount
            print "TEXT WAS: ", text
            print "COMMANDS WERE: ", commands
            print traceback.print_exc()
            _,_,tb = sys.exc_info()
            traceback.print_tb(tb)
            print "==================="
    def __buildFeatureList__(self):
        """
        internal method for building a list of all features.
        """
```

```
        for item in self.dataset:
            for key in item.keys():
                if not key in self.featureList:
                    self.featureList.append(key)
    def writeARFFfile(self,filename='data.arff'):
        """
        writes ARFF data to file
        Do not run until all data has been generated
        Using the loadTextData method or the loadFile method
        """
        data = self.createARFF()
        arfffile = open(filename,'w')
        arfffile.writelines(data)
        arfffile.close()
    def createARFF(self):
        """
        returns a arff format string
        Do not run until all data has been generated
        This is intended as an internal method
        use createARFF method instead"""
        self.__buildFeatureList__()
        string = self.getArffHeader()
```

```
        string += "@data\n"
        count = 1
        for item in self.dataset:
            count +=1
            string += self.getArffItem(item, 'arffsparse')
        return string
    def getArffHeader(self):
        """
        returns string for arff header -
        do not run until all data has been generated
        internal method for ARFF data generation
        """
        string = "@RELATION dataset\n"
        string += "\n\n"
        #string +='@ATTRIBUTE dummystring STRING\n'
        classtypes = []
        for item in self.featureList:
            if item == 'inputText':
                 string += "@ATTRIBUTE " + item + ' ' + 'STRING\n'
            elif item != 'classType':
                item = item.replace(',','CM')
                item = item.replace('"','LDQ')
```

```
                item = item.replace("'",'LQ')

                string += "@ATTRIBUTE " + item.replace(',','CM') +
                ' ' + 'NUMERIC\n'

        for item in self.dataset:

            if not item['classType'] in classtypes:

                classtypes.append(item['classType'])

        string += "@ATTRIBUTE class {"

        for item in classtypes:

            string += item + ","

        string = string[:-1] +"}"

        string += '\n\n'

        return string

    def getArffItem(self, fdict = {}, format='arffsparse'):

        """ internal method for generating an individual weka
        format data feature set from loaded data - do not run until
        data is loaded

        """

        string =""

        if format == 'arffsparse':

            string +="{" #string +="{0 'dummyvalue',"

        tuples = []

        ARFFfeatureList = []

        ARFFfeatureList = self.featureList if 'classType' in
        ARFFfeatureList:
```

```
    ARFFfeatureList.remove('classType')
for key in fdict.keys():
    if not key == 'classType':
        # we use the key to get
        # the index number for the data point
        index = ARFFfeatureList.index(key)
        # we create a tuple from the index,
        datapoint = str(fdict[key])
        datapoint = datapoint.replace(',',' CM')
        datapoint = datapoint.replace('"',' DQ ')
        datapoint= datapoint.replace("'",' SQ ')
        tuples.append(((index),datapoint,key))
        #print index, fdict[key].replace(',','CM'), key
tuples.sort()
for tup in tuples:
    if tup[2] == 'inputText':
        string += str(tup[0])+ ' "' + tup[1] + '",'
    elif not tup[2] == 'classType':
        if not float(tup[1]) == 0:
            string += str(tup[0])+ ' ' + tup[1] + ','
string      +=      str(len(ARFFfeatureList))      +      '
"'+fdict['classType']+'"'
```

```
    string +="}\n"
elif format == 'arff':
    pass
return string
```

Acknowledgements

Our research would not have been possible without access to soft- ware packages made freely available by other researchers and individuals in particular the NLTK Natural Language Toolkit, the Weka Machine Learning Package and Montylingua. We gratefully acknowledge the work of these researchers. We also thank the reviewers whose suggestions have assisted us in improving this final version of the paper.

References

Eloise Abrahams (2003), *Efficacy of plain language drafting in labour legislation*. Master's thesis on Human Resource Management), Cape Peninsula University of Technology, South Africa.

Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton (2010), *Readability assessment for text simplification*. In Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, pp. 1-9.

F.A.R. Bennion (1983), *Statute law*. Oyez

Robert W. Benson (1984), *End of legalese: The game is over*. NYU Review of Law & Social Change , Vol. 13, p. 519.

Steven Bird, Edward Loper, and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.

J.R. Bormuth (1967), *Cloze readability procedure*. University of California Los Angeles.

Kevyn Collins-Thompson and James P Callan (2004), *A language modeling approach to predicting reading difficulty.* In HLT-NAACL, pp. 193-200.

O. De Clercq, V. Hoste, B. Desmet, P. Van Oosten, M. De Cock, and L. Macken (2013), *Using the crowd for readability prediction*. Natural Language Engineering, pp. 1-33.

Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi (2011), *Read-it: Assessing readability of Italian texts with a view to text simplification.* In Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies, Association for Computational Linguistics, pp. 73-83.

W.H. DuBay (2004), *The principles of readability.* Impact Information, pp. 1-76.

Lijun Feng, Martin Jansche, Matt Huenerfauth, and Noémie Elhadad (2010), *A comparison of features for automatic readability assessment.* In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Association for Computational Linguistics, pp. 276-284.

W.N. Francis and H. Kucera (1964), *A Standard Corpus of Present-Day Edited American*. Revised 1971, Revised and Amplified 1979. Department of Linguistics, Brown University Providence, Rhode Island, USA. Available at: www.hit.uib.no/icame/brown/bcm.html (accessed 10th December, 2013)

GLPi and V. Smolenka (2000), *A Report on the Results of Usability Testing Research on Plain Language* Draft Sections of the Employment Insurance Act. Available at:
http://www.davidberman.com/wp-content/uploads/glpi-english.pdf
(accessed 10th December, 2013)

Mark Hall, Eibe Frank, Geofrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten (2009). *The weka data mining software.* ACM SIGKDD Explorations, Vol. 11, No. 1.

J. Harrison and M. McLaren (1999), *A plain language study: Do New Zealand consumers get a "fair go" with regard to accessible consumer legislation*. Issues in Writing, Vol. 9, pp. 139-184.

Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi (2008), *An analysis of statistical models and features for reading difficulty prediction*. In Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, pp. 71-79.

P. Heydari and A.M. Riazi (2012). *Readability of texts: Human evaluation versus computer index.* Mediterranean Journal of Social Sciences, Vol. 3 No. 1, 2012, pp. 177-190.

Miller J. (2005), *The development of the legal information institutes around the world*. Canandian Law Library Review, Vol. 30, No. 1, p. 8

Simon James and Ian Wallschutzky (1997), *Tax law improvement in Australia and the UK: the need for a strategy for simplification.* Fiscal Studies, Vol. 18 No. 4, pp. 445-460

Rohit J Kate, Xiaoqiang Luo, Siddharth Patwardhan, Martin Franz, Radu Florian, Raymond J Mooney, Salim Roukos, and Chris Welty (2010). *Learning to predict readability using diverse linguistic features.* In Proceedings of the 23rd International Conference on Computa- tional Linguistics, Association for Computational Linguistics, pp. 546-554

J. Kimble (1994), *Answering the critics of plain language*. The Scribes Journal of Legal Writing, Vol. 5, p. 51.

G.R. Klare (2000), *Readable computer documentation.* ACM Journal of Computer Documentation (JCD), Vol. 24, No. 3, pp. 148-168

Uta Kohl (2005), *Ignorance is no defense, but is inaccessibility? On the accessibility of national laws to foreign online publishers.* Information & Communications Technology Law, Vol. 14, No. 1, pp. 25-41

Hugo Liu (2004), *Montylingua: An end-to-end natural language processor with common sense.* Available at: http://web.media.mit.edu/~hugo/montylingua/ (accessed 10th December, 2013)

P.W. Martin (2000), *The mushrooming virtual law library on the net*. In Cornell Law Forum, Vol. 27.

D. Melham (1993), *Clearer Commonwealth Law: Report of the Inquiry into Legislative Drafting by the Commonwealth*. Technical report, House of Representatives Standing Committee on Legal and Constitutional Affairs.

Jay Milbrandt and Mark Reinhardt (2012), Access Denied: Does Withholding the Law Violate Human Rights? Regent Journal of International Law, *Forthcoming.* Available at SSRN: http://ssrn.com/abstract=2132672 (accessed 10th December, 2013)

Robert Munro, Steven Bethard, Victor Kuperman, Vicky Tzuyin Lai, Robin Melnick, Christopher Potts, Tyler Schnoebelen, and Harry Tily (2010), *Crowdsourcing and language studies: the new generation of linguistic data.* In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, Association for Computational Linguistics, pp. 122-130.

PCO NZ (2007), *Presentation of New Zealand Statute Law*: Issues Paper 2. Technical Report 2, New Zealand Law Reform Commission and New Zealand Parliamentary Counsel's Office.

PCO NZ (2008), *Presentation of New Zealand Statute Law*. Technical Report 104, New Zealand Law Reform Commission and New Zealand Parliamentary Counsel's Office.

OLR (2003), *Inland Revenue Evaluation of the Capital Allowances Act 2001* rewrite, Opinion Leader Research. Technical report, UK Inland Revenue.

OPC-Australia (2003), *Plain English*. Technical report, Australian Common wealth Office of Parliamentary Counsel.

OPC-UK (2013), *When Laws Become Too Complex: A Review into the Causes of Complex Legislation.* Technical report, United Kingdom Office of Parliamentary Counsel.

PCO-NZ (2011). *A Review of Methods for Measuring the Quality of Legislation.* Technical report, New Zealand Parliamentary Counsel's Office.

N. Pettigrew, S. Hall, and D. Craig (2006), *The Income Tax (Earnings and Pensions) Act* - Post-Implementation Review, Final Report MORI.

Emily Pitler and Ani Nenkova (2008), *Revisiting readability: A unified framework for predicting text quality*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 186-195.

G. Richardson and D. Smith (2002). *Readability of Australia's goods and services tax legislation: An empirical investigation*, Federal Law Review, Vol. 30, p. 475.

Adrian Sawyer (2010), *Enhancing compliance through improved readability: Evidence from New Zealand rewrite experiment*. Recent Research on Tax Administration and Compliance.

Sarah E Schwarm and Mari Ostendorf (2005), *Reading level assessment using support vector machines and statistical language models*. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 523-530

Luo Si and Jamie Callan (2001), *A statistical model for scientific readability.* In Proceedings of the tenth international conference on Information and knowledge management, ACM, pp. 574-576.

Johan Sjöholm (2012), *Probability as readability: A new machine learning approach to readability assessment for written Swedish*. PhD thesis, Linköpings University, Sweden. Available at: http://www.ida.liu.se/projects/webblattlast/Rapporter/lasbarhet.pdf (accessed 10th December, 2013)

D. Smith and G. Richardson (1999), *The readability of Australia's taxation laws and supplementary materials: an empirical investigation.* Fiscal Studies, Vol. 20, No. 3, pp. 321-349.

Edwin Tanner (2002), *Seventeen years on: Is Victorian legislation less grammatically complicated*. Monash University Law Review, Vol. 28, p. 403.

C van Noortwijk, RV De Mulder, and RW van Kralingen (1995), *Word use in legal texts: statistical facts and practical applicability*. Legal Knowledge Based Systems: Telecommunication and AI & Law (JURIX95), Lelystad: Koninklijke Vermande, pp. 91-100.

G. Venturi (2008), *Parsing legal texts. A contrastive study with a view to Knowledge Management Applications*. In Language Resources and Evaluation LREC 2008 Workshop on the Semantic Processing of Legal Texts, p. 1.

G. Wagner (1986), *Interpreting cloze scores in the assessment of text readability and reading comprehension.*

B. Woods, G. Moscardo, T. Greenwood, et al. (1998), *A critical review of readability and comprehensibility tests.* Journal of Tourism Studies, Vol. 9, No. 2, pp. 49-61