# LEXA: Towards Automatic Legal Citation Classification

Filippo Galgani and Achim Hoffmann

School of Computer Science and Engineering
The University of New South Wales, Sydney, Australia
{galganif,achim}@cse.unsw.edu.au

**Abstract.** In this paper we present our approach towards legal citation classification using incremental knowledge acquisition. This forms a part of our more ambitious goal of automatic legal text summarization. We created a large training and test corpus from court decision reports in Australia. We showed that, within less than a week, it is possible to develop a good quality knowledge base which considerably outperforms a baseline Machine Learning approach. We note that the problem of legal citation classification allows the use of Machine Learning as classified training data is available. For other subproblems of legal text summarization this is unlikely to be the case.

## 1 Introduction

The legal field is strongly affected by the problem of information overload, due to the large amount of legal material stored in textual form. Past decisions can have a binding effect on following decisions, in a process that is known as *stare decisis* [12], especially in countries with common law systems, such as Australia, UK and USA. As a consequence, judges need to know past cases to be coherent and "just" in their application of law and lawyers use them to find arguments for their cases. Court decisions or cases can be instructive as they introduce a new principle or rule, modify or interpret an existing principle or rule, or settle a question upon which the law is doubtful.

A number of different approaches of information management from other domains have been carried over to the legal domain: for example automatic summarization [9,6], retrieval [11] and information extraction [16]. However, researchers in the field, such as Hachey and Grover, already noted:

> "[...] *the legal domain appears to be more complex than scientific articles and especially news, the most commonly reported domains in the automatic summarization literature. This is evidenced in characteristics of legal discourse such as the longer average sentence lengths, longer average document lengths, and the sometimes convoluted and philosophical nature of legalese where there is not an absolute logical template and there is a looser notion of topic which lends itself to a less centralized focus*" [8].

J. Li (Ed.): AI 2010, LNAI 6464, pp. 445–454, 2010.

Automatic summarization of legal cases can support finding (helping in assessing the relevance of the results of a query) and digesting the right documents. Furthermore it can aid the manual creation of summaries and provide important legal information in a format that is more accessible and understandable. In case-law systems, such as in Australia, because of the importance of relying on other cases to answer the case at hand, citations are an important aspect of most judicial decisions. Examining citations tells us how the law we are relying on has been interpreted. For this reason it is vital to law professionals to know whether the decision has received positive, negative, cautionary or neutral treatment in subsequent judgements.

In this paper we outline our system LEXA (Legal tEXt Analyzer): an approach towards automatically providing information useful to law professionals from such case reports. LEXA is based on incremental acquisition of annotation rules, and we describe an evaluation of it on a citation classification task.

The following section discusses related work. This is followed by the description of our annotated legal corpus in Section 3. In Section 4, we present our approach towards legal citation classification. The following Section 5 discusses our first results on legal citation classification. The final Section 6 discusses our achievements so far and outlines future research.

## 2 Related and Prior Work

In the past automatic summarization has attracted a large body of research, and a large variety of techniques and approaches have been proposed for this task. Although there has been a certain amount of research in summarization of legal texts, this application domain is not mature as other such as news or scientific articles. Examples of systems for automatic summarization of legal texts are LETSUM [6] and the work of Hachey and Grover [9].

To our knowledge there have been no attempt to automatically classify citations in legal cases, citation classification has been applied mainly in the domain of scientific papers. Following the pioneering approach of Nanba and Okomura [15], another system that automatically perform citation classification is described in [20], where different kinds of features are studied to train an IBk classifier on an annotated corpora.

In 2004 Nakov et.al. [14], pointed out the possibility to use citations contexts directly for text summarization, as they provide information on the important facts contained in the paper. A first application of the idea can be found in the work of Qazvinian and Radev [18], where they create a summary by extracting a subset of the sentences that constitute the citation context. Mohammad et.al. [13] apply this approach to multi-document summarization, they also build up on the claim by Elkiss et.al. [5] about the difference of information given by the abstract and the citation summary of a paper.

Our knowledge acquisition (KA) methodology is based on incremental approaches, in particular on the Ripple Down Rules (RDR) methodology [3]. In RDR, rules are created manually by domain experts without a knowledge engineer. The knowledge base is built with incremental refinements from scratch, while the system is running: a domain expert monitors the system and whenever it performs incorrectly, he signals the error and provides a rule as a correction.

A Single Classification RDR (SCRDR), see Figure 1, is a binary tree; associated with each node is a rule (a condition and a conclusion). Cases (objects to be classified) are evaluated as they are passed from node to node, starting from the root: if the condition of the node is satisfied we follow the so-called *except* edge (we say that the node *fires*), otherwise the *if not* edge, if there is any. The final conclusion given by the SCRDR tree is the conclusion of the node that fired last, i.e. that is deepest in the tree (but is often not a leaf node). To ensure that a conclusion is always found, the root node typically contains a trivial condition which is always satisfied. This node is called the *default* node. When an instance is misclassified, a new node is added to the tree. If the node $n_f$ that fired last has no except link, a new except link is created and the new node is attached to it. If $n_f$ has already an except link leading to node $n_e$ the new node is attached as an alternative except link. That is done by following the *if-not* link chain of $n_e$ until no *if-not* link is found and then creating a new *if-not* link and attaching the new node to that link. Then the domain expert formulates a rule for the new node that is satisfied by the case. This rule represents an explanation for why the conclusion on the case at hand should be different. The strength of RDR is easy maintenance: the point of failure is automatically identified, the expert patches the knowledge only locally, considering the case at hand, and new rules are placed by the system in the correct position and checked for consistency with all cases previously correctly classified.

RDR have been applied to different problems and applications. For a recent survey see [19]. RDR has also been extended to tackle natural language processing tasks. Among such work is also the work on scientific citation classification in [17] on which the work in this paper builds to a significant extent. However, our application domain is considerably more complex. Hence, this paper demonstrates that the Ripple Down Rules approach also successfully extends to more complex NLP domains.

## 3 Creating Our Corpus of Legal Citations

AustLII (the Australasian Legal Information Institute) [1,7] provides free access to a large amount of legal information, including reports on court decisions in all major courts in Australia. A similar project, the World Legal Information Institute (WordLII), is an extension to other countries, with the aim of providing "*free, independent and non-profit access to worldwide law*".

We accessed the court case reports in html format from the AustLII website. Some of the contained citations are marked up with a hyperlink to the corresponding cited case. Notably, some of the Federal Court of Australia (FCA) cases also contain expert generated citation classes of the cited cases.

We built a robust parser to analyse the html pages of the FCA reports and extract the relevant information about the citations.

Examples of the classified citations in FCA documents are:

- Dunstan v Human Rights and Equal Opportunity Commission (No 2) [2005] FCA 1885 related
- Australian Fisheries Management Authority v PW Adams Pty Ltd (No 2) (1996) 66 FCR 349 distinguished
- Copping v ANZ McCaughan Ltd (1997) 67 SASR 525 cited
- DJL v Central Authority [2000] HCA 17; (2000) 201 CLR 226 considered

We can decompose each row in the name (e.g. DJL v Central Authority), the legal citations (e.g. [2000] HCA 17) and the class (e.g. considered). The distribution among the citation classes for the years 2007-2009 from 2043 FCA documents containing 18715 labelled citations is shown in Table 1.

**Table 1.** Distribution of citation classes for 2007-2009. Those selected for our knowledge acquisition task are in bold.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Cited | 9346 | Referred to | 3017 | **Applied** | **1803** | **Followed** | **1759** |
| Considered | 1339 | Discussed | 706 | **Distinguished** | **463** | Related | 94 |
| Affirmed | 91 | Quoted | 87 | Approved | 61 | Not Followed | 57 |
| Reversed | 20 | Ref to | 15 | Explained | 10 | Questioned | 9 |
| Disapproved | 8 | Noted | 7 | Relied on | 4 | Doubted | 3 |
| Compared | 2 | Adopted | 2 | Overruled | 2 | Referred | 2 |

It is possible that a case is cited differently, in different citing cases, or even within the same citing case, due to the fact that different aspects of the cited case may be of interest. As a consequence, combinations such as Applied/Distinguished are possible and one citation may have more than one class label attached, though this is rare.

Finding all occurrences of a citation in a case is not trivial, as references are made in a large variety of ways as opposed to scientific articles. Different ways of referring to the same case include:

- The full name of the case, e.g. Yevad Products Pty Ltd v Brookfield [2005] FCAFC 263; (2005) 147 FCR 282, or just Yevad Products Pty Ltd v Brookfield
- The name of one of the parts, e.g. Yevad or Brookfield
- Indication of the law report, e.g. (2005) 147 FCR 282, or the medium neutral citation, e.g. [2005] FCAFC 263
- Combination of these components, for example Brookfield 147 FCR
- The name of the respective judge, e.g. *In Burgundy Royale Brennan J at 685 said:* or *I understand Brennan J's reference to the prospect of a grant of special leave...*

When resolving the last type of reference we only use those citations that are unambiguous, e.g. where a judge's name is only involved in one possible cited case. To turn this information into a training corpus usable for supervised machine learning, we attached to each citation its class label and the associated sentence(s) as well as the entire paragraph(s) in which the citation occurs. Where a citation is mentioned multiple times we collect the sentences (paragraphs) surrounding each of the occurrences. For our study we only used a subset of our training corpus as described in Section 5, but we are going to make the entire corpus available for other researchers.

## 4 Building Legal Citation Classification Systems

Our citation classification system is based on creating a knowledge base of rules to annotate text, and in particular to annotate citations. We built an application based on GATE [4]. GATE is a framework for developing components for processing human language including text, written in Java and available as free open-source software. We use the Tokenizer, Sentence Splitter, Part of Speech Tagger and Stemmer resources (provided with GATE) to generate Token annotations and their corresponding features for input texts, building the first layer of linguistic annotations.

Our system LEXA for classification of citations is based on customized annotations produced by a knowledge base of rules, which take the form of regular expressions over annotations. Each rule matches a regular expression of tokens and other annotations, and posts a new annotation. To create these annotations we used the Semantic Tagger from GATE, a finite state transducer which annotates text based on JAPE (Java Annotation Patterns Engine) grammars.

Our system is composed of different types of rules. A first group of rules is used in a preliminary phase to identify judge names, parts (e.g. "the plaintiff", "the appellant"...), courts (e.g. "a full court of the HCA"...), citations of paragraphs (e.g. "case at [145-148]" ), etc. The user can create any type of annotation as needed. A successive level of annotation aims at posting class labels over citations. This class of rules posts a particular annotation which specify which class we believe the citation belongs to.

This set of classification rules is acquired interacting with a human user. The system displays citations (and surrounding text) to the user, specifying the (known) type of citation. The user, examining the text at hand, can create a new rule and test it on the entire corpus. At the first level we created rules to extract Distinguished citations. The user is presented with the surrounding text of a Distinguished case not yet classified, and creates a rule to identify that case. The rule can be tested on the entire corpus to see how many Distinguished and Followed/Applied cases it matches. When the user is satisfied with the rule, he/she commits the rule to the knowledge base.

When a new case of class Followed/Applied is considered and incorrectly classified as Distinguished by a first level rule, the user will add a second level exception rule reverting the tentative classification of Distinguished to Followed/Applied.

After the knowledge base is built, the classification of new citation cases is done as described in Section 2. An example of a portion of the resulting RDR tree, including three rules, is shown in Figure 1. We conducted a number of

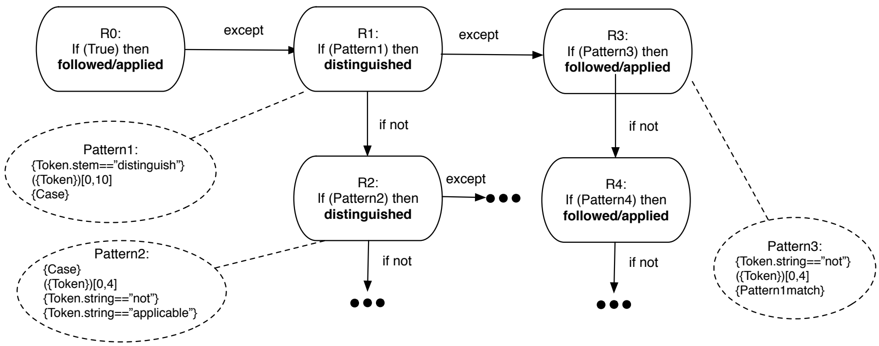


**Fig. 1.** Example of a portion of the RDR tree: R1 matches a word whose stem is "distinguish", followed by an annotation of type Case (the annotation that signal the cases in our corpus), separated by a gap up to 10 words. The exception rule R3 looks if there is a token "not" up to 4 tokens before the annotation posted by R1.

knowledge acquisition sessions (no legal expert was involved) in order to build a knowledge base with a reasonable performance on our legal citation classification corpus.

## 5 Experimental Results

We considered a two class problem: Distinguished (D) vs. Followed or Applied (FA). We believe that these classes are particularly relevant for legal professionals as they are likely to shed light on what criteria have been used to decide if a given case constitutes a relevant precedent (for Followed or Applied). Similarly, Distinguished cases would indicate why a cited case is not relevant for the citing case. The importance of these three classes is further corroborated by the fact that alternative classification schemes, e.g. from private publishers such as Lexis Nexis, coincide on those classes while they differ on many other classes. The Lexis Nexis CaseBase Case Citator [2] explains these classes as follows:

- Applied: A principle of law articulated in the primary case is applied to a new set of facts by the court in the subsequent case.
- Followed: The annotation is similar to applied but is used in circumstances where the facts in the primary case resemble reasonably closely the facts in the subsequent consideration case.
- Distinguished: The court in the subsequent case holds that the legal principles articulated by the primary case (usually otherwise persuasive or binding authority) do not apply because of some difference between the two cases in fact or law.

The respective sub-corpus for the three classes we used contains 460 Distinguished citations and 3496 Followed or Applied citations with a total of 3956 citations.

Using the approach sketched in Section 4, we developed a first knowledge base taking around 30 hours of knowledge acquisition sessions. The knowledge base contains a total of 34 rules of which 15 are second-level exception rules. Only seven of the first layer rules contain one (or more) exception rules. The time to acquire a rule is divided into reading and understanding the text and time to decide which rule should be created. The first is the most demanding part, as sometimes the text is long and complex. Sometimes it is difficult to understand why a certain class label was given.

### Comparing Our KA Approach with Machine Learning

While developing our knowledge base, we could test the performance of the rules directly on the corpus. The root node of the RDR tree classify every case as FA, i.e. it misclassifies 460 D cases. After adding exceptions rules to the default node, we recognized 207 citations of class D correctly. However also 133 cases were incorrectly classified as D. In order to rectify that we added exception rules to the first level of exception rules classifying cases as D. Those second level exception rules recognize correctly 80 out of the 133 FA cases. This second level of exceptions rules also caused 15 out of the 207 D cases to be misclassified as FA.

The knowledge base we developed in some 30 hours of knowledge acquisition. To create the 34 rules, we looked at approximately 60 cases (but we tested them on all cases before deciding to commit a formulated rule). Some of the rules were dismissed after testing and then manually refined.

To compare the performance of our knowledge base with a baseline machine learner we trained the Naive Bayes classifier in WEKA [10] using a simple bag of words model, only indicating presence or absence of a word. To build the model, for each citation we extract all the words that appear in the surrounding context (either sentence or paragraph). Comparison with alternative machine learning approaches is left for future research.

Due to the fact that NB did not do well on recognizing the minority class of Distinguished cases (which we believe is the more important one for legal practitioners) we tried to improve the Naive Bayes performance by giving more weight to the minority class, i.e. by replicating the instances of type D, with factors of 2, 4, 8, 12 and 20. We found the factors 2 and 4 to produce the best results. The results are presented in Table 2. Six Naive Bayes models are shown, using only the words present in the same sentence (s) or all the words in the paragraph (p), with the original instances or multiplying factor of 2 and 4.

We built a test corpus of unseen data by downloading all FCA cases for 2006 containing a total of 6541 citations with 1274 FA and 160 D citations. This data had never been used at any stage in developing our system. We applied our knowledge base as well as the six trained Naive Bayes classifiers to the new test data. The results are shown in Table 2.

It should be noted, however, that the human expert provided class labels are not necessarily agreed upon by other human experts. This results effectively in noise in our data for the purpose of training and testing our classifiers.

**Table 2.** NB2s: stands for Naive Bayes with words from surrounding sentence supplied and cases from class D provided twice - other column titles analogously

| | Training data 2007-2009 | | | | | | | Test data 2006 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LEXA | NBs | NB2s | NB4s | NBp | NB2p | NB4p | LEXA | NBs | NB2s | NB4s | NBp | NB2p | NB4p |
| Precision(D) | 0.784 | 0.632 | 0.814 | 0.824 | 0.806 | 0.784 | 0.208 | 0.5 | 0.137 | 0.303 | 0.493 | 0.219 | 0.359 | 0.167 |
| Recall (D) | 0.417 | 0.25 | 0.485 | 0.772 | 0.172 | 0.898 | 1 | 0.263 | 0.1 | 0.188 | 0.206 | 0.088 | 0.144 | 0.606 |
| F-measure(D) | 0.545 | 0.358 | 0.608 | 0.797 | 0.283 | 0.837 | 0.344 | 0.344 | 0.116 | 0.232 | 0.291 | 0.125 | 0.205 | 0.262 |
| Precision(FA) | 0.928 | 0.91 | 0.937 | 0.971 | 0.903 | 0.987 | 1 | 0.913 | 0.891 | 0.903 | 0.907 | 0.894 | 0.9 | 0.926 |
| Recall (FA) | 0.985 | 0.981 | 0.986 | 0.979 | 0.995 | 0.968 | 0.506 | 0.967 | 0.921 | 0.946 | 0.973 | 0.961 | 0.968 | 0.622 |
| F-measure(FA) | 0.955 | 0.944 | 0.96 | 0.975 | 0.946 | 0.977 | 0.672 | 0.939 | 0.906 | 0.924 | 0.939 | 0.926 | 0.933 | 0.744 |
| Accuracy | 0.919 | 0.897 | 0.928 | 0.955 | 0.9 | 0.96 | 0.562 | 0.888 | 0.829 | 0.861 | 0.888 | 0.864 | 0.876 | 0.62 |

**Table 3.** Removing cases with differing human expert classifications

| | Training data 2007–2009 | | | | | | | Test data 2006 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LEXA | NBs | NB2s | NB4s | NBp | NB2p | NB4p | LEXA | NBs | NB2s | NB4s | NBp | NB2p | NB4p |
| Precision(D) | 0.8 | 0.545 | 0.804 | 0.911 | 0.875 | 0.912 | 0.463 | 0.674 | 0.154 | 0.341 | 0.563 | 0.214 | 0.64 | 0.362 |
| Recall (D) | 0.516 | 0.293 | 0.511 | 0.684 | 0.187 | 0.827 | 0.987 | 0.403 | 0.083 | 0.194 | 0.25 | 0.083 | 0.222 | 0.694 |
| F-measure(D) | 0.627 | 0.382 | 0.625 | 0.782 | 0.308 | 0.867 | 0.63 | 0.504 | 0.108 | 0.248 | 0.346 | 0.12 | 0.33 | 0.476 |
| Precision(FA) | 0.903 | 0.863 | 0.904 | 0.936 | 0.852 | 0.964 | 0.996 | 0.850 | 0.772 | 0.799 | 0.818 | 0.781 | 0.816 | 0.885 |
| Recall (FA) | 0.972 | 0.948 | 0.974 | 0.986 | 0.994 | 0.983 | 0.757 | 0.946 | 0.872 | 0.895 | 0.946 | 0.914 | 0.965 | 0.658 |
| F-measure(FA) | 0.936 | 0.904 | 0.937 | 0.96 | 0.918 | 0.973 | 0.86 | 0.895 | 0.819 | 0.844 | 0.877 | 0.842 | 0.884 | 0.754 |
| Accuracy | 0.891 | 0.833 | 0.893 | 0.933 | 0.853 | 0.956 | 0.797 | 0.827 | 0.699 | 0.742 | 0.793 | 0.732 | 0.802 | 0.666 |

To identify those "noisy" or at least questionable citations, we compared the FCA citation classifications with the available classifications of the same citations by the Lexis Nexis CaseBase Case Citator [2], a commercial database of case law. In this database experts classify each citation in a scheme of eleven classes (different from the one of FCA but comparable to it). Of our 460 FCA cases marked as Distinguished, 225 of them had the same label in CaseBase. Of the 3496 Followed or Applied, only 1041 received either Followed or Applied in CaseBase.

As a consequence of this considerable discrepancy in human expert opinion, which we think is due to the class boundaries not being very sharp, it appears to be more appropriate to limit the performance evaluation in our study to those cases where both human-provided class labels (from FCA and Lexis Nexis CaseBase) agree. Results for the training and test sets (329 citations) containing only the cases the human experts agreed on, are shown in Table 3.

These results indicate that the human intuition that went into the knowledge base of our system LEXA generalises significantly better than the Machine Learner which appears to be overfitting the training data to a much higher degree than LEXA (the performance difference between training and test data is less for LEXA), with our system obtaining an accuracy of 82.7%. As a consequence, on the less ambiguous sub-corpus our knowledge base is outperforming the best version of our trained Naive Bayes classifier by a margin of up to some 45% relative to the Naive Bayes F-measure of 34.6% for the important D class.

## 6 Conclusions and Future Work

In this paper we presented our approach towards automatic legal citation classification, which characterizes the relation between the present case and the cited ones. Automatic classification of citations to case law is a novel application in itself, which we believe to be very relevant in assisting legal research in common law. Moreover, we believe that citation analysis in legal cases can bring benefits to a range of other NLP applications, including automatic summarization. We built an annotated corpus of citations, extracting available data from the Federal Court of Australia reports, which we are going to release to interested researchers.

Our system LEXA is based on a knowledge base of rules (described by regular expressions) to annotate text at multiple levels. We built a knowledge base manually acquiring a set of rules: within less than a week of knowledge acquisition sessions, our system is able to recognize Distinguished and Followed or Applied citations with an accuracy of 88.8% on test data. When evaluated on unseen data, the system outperforms our best Machine Learning model, giving significantly higher recall and precision for the Distinguished class (34.4% vs 29.1% F-measure).

Examining an alternative source of classification of the same cases, we found that human experts often disagree when classifying citations (only 1266 of the 3956 citations have the same label in both sources). This confirms the complexity of citation classification in the legal domain. We experimented that, when taking out "ambiguous" citations (the ones on which human experts do not agree), the performance increases for our knowledge base, with our system bringing a considerable improvement of F-measure both for Distinguished (50.4% vs. 34.6%) and Followed/Applied (89.5% vs. 87.7%) over Naive Bayes.

Future work involves integrating this approach in a more comprehensive analysis of the legal texts, with the aim to build an automatic summarization system. In order to achieve this, more rules of different types will be needed, to allow deeper analysis of the text. For automatic summarization the problems in obtaining annotated data makes it more difficult to use Machine Learning approaches. For this reason we think that knowledge acquisition from experts with RDR is the path to follow to obtain better results.

## References


1. Australasian Legal Information Institute, `http://www.austlii.edu.au/`
2. Lexis Nexis CaseBase Case Citator, `http://www.lexisnexis.com.au/`
3. Compton, P., Jansen, R.: Knowledge in context: a strategy for expert system maintenance. In: AI 1988: Proceedings of the second Australian Joint Conference on Artificial Intelligence, pp. 292–306. Springer, New York (1990)
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia (July 2002)
5. Elkiss, A., Shen, S., Fader, A., Erkan, G., States, D., Radev, D.: Blind men and elephants: What do citation summaries tell us about a research article? J. Am. Soc. Inf. Sci. Technol. 59(1), 51–62 (2008)
6. Farzindar, A., Lapalme, G.: Letsum, an automatic legal text summarizing system. In: Legal Knowledge and Information Systems: JURIX 2004, the Seventeenth Annual Conference, p. 11. Ios Pr. Inc., Amsterdam (2004)

7. Greenleaf, G., Mowbray, A., King, G., Van Dijk, P.: Public Access to Law via Internet: The Australian Legal Information Institute. Journal of Law and Information Science 6, 49 (1995)
8. Hachey, B., Grover, C.: Automatic legal text summarisation: experiments with summary structuring. In: ICAIL 2005: Proceedings of the 10th International Conference on Artificial Intelligence and law, pp. 75–84. ACM, New York (2005)
9. Hachey, B., Grover, C.: Extractive summarisation of legal texts. Artif. Intell. Law 14(4), 305–345 (2006)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
11. Moens, M.F.: Innovative techniques for legal text retrieval. Artificial Intelligence and Law 9(1), 29–57 (2001)
12. Moens, M.F.: Summarizing court decisions. Inf. Process. Manage. 43(6), 1748–1764 (2007)
13. Mohammad, S., Dorr, B., Egan, M., Hassan, A., Muthukrishan, P., Qazvinian, V., Radev, D., Zajic, D.: Using citations to generate surveys of scientific paradigms. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, pp. 584–592 (June 2009)
14. Nakov, P.I., Schwartz, A.S., Hearst, M.A.: Citances: Citation sentences for semantic analysis of bioscience text. In: Proceedings of the SIGIR 2004 Workshop on Search and Discovery in Bioinformatics (2004)
15. Nanba, H., Okumura, M.: Towards multi-paper summarization using reference information. In: IJCAI 1999: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 926–931. Morgan Kaufmann Publishers Inc., San Francisco (1999)
16. Palau, R.M., Moens, M.F.: Argumentation mining: the detection, classification and structure of arguments in text. In: ICAIL 2009: Proceedings of the 12th International Conference on Artificial Intelligence and Law, pp. 98–107. ACM, New York (2009)
17. Pham, S.B., Hoffmann, A.: A new approach for scientific citation classification using cue phrases. In: Proceedings of Australian Joint Conference in Artificial Intelligence (2003)
18. Qazvinian, V., Radev, D.R.: Scientific Paper Summarization Using Citation Summary Networks. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 689–696 (2008)
19. Richards, D.: Two decades of Ripple Down Rules research. Knowl. Eng. Rev. 24(2), 159–184 (2009)
20. Teufel, S., Siddharthan, A., Tidhar, D.: Automatic classification of citation function. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 103–110. Association for Computational Linguistics, Sydney (July 2006)