How Machine Learning can be Beneficial for Textual Case-Based Reasoning

Stefanie Brüninghaus and Kevin D. Ashley
University of Pittsburgh
Learning Research and Development Center, Intelligent Systems Program, and School of Law
3939 O'Hara Street, Pittsburgh, PA 15260
steffi+@pitt.edu, ashley+@pitt.edu

Abstract

In this paper, we discuss the benefits and limitations of Machine Learning (ML) for Case-Based Reasoning (CBR) in domains where the cases are text documents. In textual CBR, the bottleneck is often indexing new cases. While ML has the potential to help build large case-bases from a small start-up collection by learning to classify texts under the index-terms, we found in experiments with a real CBR system, that the problem is often beyond the power of purely inductive ML. CBR indices are very complex and the number of training instances in a typical case base is too small reliably to generalize from. We argue that adding domain knowledge can help overcome these problems and give illustrating examples.

CBR over Textual Cases

Case-Based Reasoning has been successfully applied in various domains where the cases are available as text documents, e.g., Legal Reasoning and Argumentation (Aleven 1997), Ethical Dilemmas, Medical Applications, Tutoring, or Helpdesk systems. Up to now, the case bases for systems in these domains had to be constructed by hand. This leads to a bottleneck in creating and scaling up CBR systems, since manual indexing often involves inhibitory costs: Candidate cases have to be retrieved, the most useful cases have to be selected, read and understood, and transcribed into the CBR system's representation. Thus, methods for automatically assigning indices or improving case representation and reasoning from texts are needed.

However, there is a severe gap between the knowledge representation required for CBR and the methods one can perform on textual documents, like in Information Retrieval (IR). Computationally, text documents are an unstructured stream of characters, over which only shallow reasoning based on easily observable surface features can be performed. This reasoning can be distinguished from CBR in many material ways. In CBR, cases are represented in a meaningful, very compact way, and not as very high-dimensional, hardly interpretable vectors of floating-point numbers over words. They usually contain only important information, noisy or irrelevant information is filtered out in the indexing process. Cases can be compared along multiple important dimensions, and partial matches, can be adapted to a problem situation, using domain knowledge contained in the system (Aleven 1997). Thus, approaches based only on shallow statistical inferences over word vectors, are not appropriate or sufficient. Instead, mechanisms for mapping textual cases onto a structured representation are required. In the following sections, we will discuss our legal CBR application, and the problems we encountered when using ML methods for deriving the mapping between textual cases and a CBR-case representation.

Application: CBR in CATO

Our particular CBR application is CATO, an intelligent learning environment for teaching skills of making arguments with cases to law students. It is implemented for the domain of trade secrets law, which is concerned with protecting intellectual property rights of inventors against competitors. In the CATO model, cases are represented in terms of 26 factors. These are prototypical fact situations which tend to strengthen or weaken the plaintiff's claim. Cases are compared in terms of factors, and similarity is determined by the inclusiveness of shared sets of factors. High-level knowledge about trade secrets law is represented in a Factor Hierarchy (see Fig. 1), where the base-level factors are linked to more high-level legal issues and concerns via a specific support mechanism. With the Factor Hierarchy, cases that match only partially can be compared in terms of abstractions, and CATO can reason context-dependently about similarities and differences between cases.

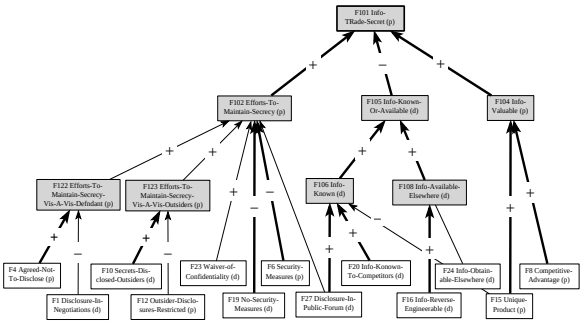

Figure 1: CATO's Factor Hierarchy

The assignment of factors to new cases can be treated as a learning problem. Each of CATO's factors corresponds to a concept, for which there are positive and negative full-text opinion instances. The learning task can be defined as:

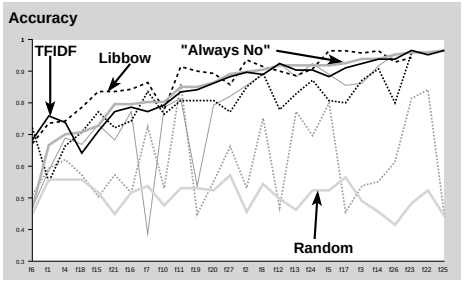

Figure 2: Results of experiments

For each of CATO's factors:

Given a set of labeled full-text legal opinions,

Learn a classifier which determines for a previously unseen opinion, whether the factor applies or not.

The classifiers for the individual factors are not equally difficult to learn. For some, we have as few as five positive instances in the Case Database, while others apply in almost half of the cases. Clearly, the factors with few positive instances are harder to induce. The factors also relate to distinct real-world situations of different character. For instance, f4, Non-Disclosure Agreement, is usually described in a more uniform way than f6, Security Measures. F6 captures a much wider variety of situations, and therefore, it is easier to derive a classifier for cases related to f4.

In order to bring the texts in a feature/value representation computationally accessible by an ML algorithm, we treated the texts as bag-of-words. All punctuation and numbers, including citations of other cases were filtered out. After stemming, we removed stop-words and very rare terms. Terms included single words, and adjacent pairs of non stop-words. For term weighting, we used tfidf. We did not consider any semantic information.

For the experiments, we implemented Rocchio, TFIDF (Joachims 1996), Winnow, Weighted Majority (Blum 1995), Exponentiated Gradient and Widrow-Hoff (Lewis *et al.* 1996), and ran the Naive Bayes classifier from the Libbow/Rainbow (McCallum 1997).

In order to assess performance, we decided to use accuracy as measure. In our application, we do not have enough cases so that the usual interpretations of precision/recall would hold. Also, the absence of a factor can be very relevant in making legal arguments, which makes the recognition of positive and negative instances of the concepts relevant. Hence, we used accuracy as the main criterion to measure performance. However, for factors with few positive instances, high accuracy can be achieved by always labeling new cases as negative. Therefore, we also consider precision/recall, to discover this undesired behavior.

The cases in the Case Database were assigned to test and training sets in 10-fold, randomized cross-validation runs. Disappointingly, in the experiments, all algorithms performed quite poorly on our data set (Brüninghaus & Ashley 1997). The results are shown in Fig. 1, where the accuracy is plotted over the factors, ordered by the number of positive instances. Only Rocchio, Libbow and TFIDF, as shown in Fig. 2, achieved acceptable results in terms of both, accuracy and precision/recall, for the factors where we had a relative large number of positive training instances. On all other factors, we often observed that the algorithms could not discover positive cases, and always labeled opinion texts as negative, so their accuracy is pretty close to the "always no" strategy. Depending on parameter settings, the online learning algorithms showed basically a random behavior. These results did not confirm our hopes raised by results reported previously for experiments on different problems. For a comprehensive overview and comparison of text classification experiments, see (Yang 1997). We think that for our problem these elsewhere successful methods did not work that well because:

- The factors are very complex concepts to be learned. A variety of circumstances and situation can influence the presence of a factor, and the information can be in multiple passages within the text. Sometimes, the evidence for a factor can be indirect. E.g., it is considered to be improper means (a factor favoring plaintiff in CATO), if defendant searches plaintiff's garbage. The reasoning underlying this factor assignment is very hard to capture in existing text classification systems without any background or semantic knowledge.
- CATO's Case Database of 150 cases is magnitudes smaller than collections like the Reuters newswire stories or the Medline data (Yang 1997). Even so, it is fairly large for a CBR system, tremendous time and effort went into it. It is unrealistic to hope for a case-base of a size comparable to the Reuters collection with thousands cases for a real CBR application, so methods that can generalize reliably from small case-bases have to be found.

Suggestions

In spite of these discouraging preliminary results, we believe ML can help overcome the indexing bottleneck. Rather than improving upon purely inductive learning methods, we think that integrating available background knowledge is the most promising way to improve performance. With few exceptions (notably (Koller & Sahami 1997)), background knowledge has not been used. The learning methods have been generic and domain-independent. CBR is application specific, knowledge about the domain and the use of cases is captured in case representation, similarity measure and adaptation mechanisms. In CATO, this knowledge is represented in the Factor Hierarchy (see Fig. 1). We have worked out a set of hypotheses concerning in what ways this knowledge can be employed to overcome the difficulties described above.

To illustrate our ideas, we will focus on an example from CATO's case database. In *Peggy Lawton*, an employee of a food company resigned after discovering his employer's secret cookie recipe. When he sold cookies baked with an identical recipe, his former employer sued him for trade secret misappropriation. In CATO's Case Database, this is represented by factors F6 (Security-Measures), F15 (Unique-Product) and F18 (Identical-Products).

HYPOTHESIS I *A knowledge-rich representation of the meaning of terms used will improve the learning process and enhance the effectiveness of a classifier.*

Typically, documents are represented in IR and ML by splitting the text into words, and taking those as tokens. However, this approach neglects the specialized language used in an area of discourse, and the meaning inherently assigned with technical terms. The highest weighted terms in *Peggy Lawton*'s document vector would be more relevant for a cookbook than for a legal application: `kitchen cooki chocolatchip chocolat hogan hogie hogiebear recip chip cookie chipcooki lawton wolf ingredient bak`.

tf/idf term weighting assigns high weights to terms that allow identifying this document within the collection, thereby overemphasizing words not useful for assigning factors.

We believe that identifying domain specific concepts in the document text, e.g., using a concept dictionary, can help more reliably to determine what the document is about by focusing ML, matching and ranking algorithms on the more relevant input tokens.

In an experiment, we limited the vocabulary manually, by removing all tokens irrelevant for trade secrets law, and ran the algorithms listed above. Similarly, we collected the West Legal Publisher's keynotes related to trade secrets law, and extracted the tokens from them, as described above. The keynote system has for every case a set of manually assigned abstract indices and short descriptions. Although this reference system is fairly general, and not useful for CBR, it greatly facilitates legal research. When the ML algorithms were only given a limited set of feature/value pairs, performance was somewhat improved. In particular for those factors with relatively many positive instances, we observed higher precision using either strategy, while accuracy and recall remained constant.

HYPOTHESIS II *Decomposing the problem by first determining which broader index applies to a case, and then refining the indexing decreases the complexity of the learning task.*

Another technique potentially useful is to decompose the problem into less complex subproblems by assigning issues first, and then use this as input to subsequent classifiers which identify more specific factors presented in the text. Intuitively, it is easier to decide which general issues a case involves than assigning the more detailed factors. Once issues have been identified, the Factor Hierarchy's relations of issues to factors can be used to focus learning on particular factors, reducing the hypothesis space.

Initial experiments suggest that learning a classifier for deciding which issues apply in a case is still not as reliable as we would hope, but the algorithms did considerably better than on the base-level factors.

This idea is similar to the hierarchical text classification suggested in (Koller & Sahami 1997). There, evidence was presented how a hierarchical classification scheme can increase performance, and at the same time decrease the number of attributes necessary to represent a document.

HYPOTHESIS III *Relations among factors and issues in the Factor Hierarchy can enhance performance by supporting predictions of the likelihood that a factor applies in situations where training instances are scarce.*

In order to compensate for the small number of instances for some of the factors, we intend to employ heuristics based on the meaning of factors and on their relationships expressed in the Factor Hierarchy. Based on either semantic or empirical grounds, one may infer that certain factors naturally occur together (or not). The Factor Hierarchy expresses semantic relations among factors via more abstract parent concepts (i.e., issues and other high-level factors), which can provide confirming evidence for a factor in a text and to test the credibility of an assignment of factors.

Consider, for instance, F23, Waiver-Of-Confidentiality, which applies to only six cases in the database. The concept is very hard to find, since it can involve diverse fact situations. It would be much easier to train a classifier to identify factors F1, Disclosure-In-Negotiations, or F21, Knew-Info-Confidential, which apply in many more cases. F23 has some useful relationships with these easier-to-learn factors. There would rarely be a waiver of confidentiality without a disclosure in negotiations. It would also be unlikely to find a waiver of confidentiality in a case where there is a non-disclosure agreement (represented by F4) or knowledge that information is confidential. Our expectations about these relationships can be expressed probabilistically, and then used with Bayes' Law to help assess whether F23 is in fact present: $\mathrm{E}(F1|F23) \mapsto high, \mathrm{E}(F4|F23) \mapsto low$.

The knowledge about the relations among factors represented in CATO's Factor Hierarchy can also be used to check the factors assigned by the system for consistency and detect incorrect classifications. Moreover, if it is possible to determine from an (Information Extraction-like) analysis of the text which side won, plaintiff or defendant, a system can also reason about whether the assigned factors are consistent with the result. If only pro-defendant factors were found, but plaintiff won, the factor assignment is suspect.

HYPOTHESIS IV *Spotting quotations of statutory texts in an opinion should help identify applicable issues.*

In legal opinion texts, the courts refer to the relevant statutes or restatement commentaries to provide a justification for their decision. Spotting these quotations is evidence for the issue raised. The following quote of the (First) Restatement of Torts § 757 can be found in *Motorola*:

> *(a) substantial element of secrecy must exist, so that, except by the use of improper means, there would be difficulty in acquiring the information. ... Matters of public knowledge ... cannot be appropriated by one as his secret.*

This code section is related to high-level factor F120 (Info-Legitimately-Obtained-Or-Obtainable), in CATO's Factor Hierarchy. Supporting evidence for F120 is high-level factor F105 Info-Is-Known-Or-Available ("general knowledge"), while Questionable-Means (F111), referred to as "improper" above, blocks this conclusion. From the quote in *Motorola* one may expect that these abstract factors are related to the issues discussed in the text of the opinion.

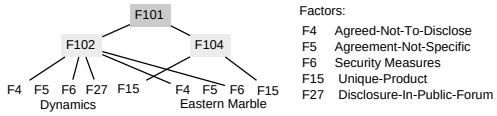

Figure 3: Citations and High-Level Factors in *Peggy Lawton*

HYPOTHESIS V *Accounting for document structure can help identify relevant text segments, thus focussing the classifier on predictive input.*

Information about the structure of the text of an opinion can be employed beneficially in identifying important and less relevant sections. While not following a rigid format, legal opinions do have a basic structure: first the facts of the case are described, then the applicable law is stated and applied to the facts, and finally, the legal consequences are drawn. If a system can detect this structure, the information contained in the different parts can be used more specifically. Paragraphs related to unrelated legal matters should be ignored, they introduce irrelevant information, or noise.

HYPOTHESIS VI *Identifying cases cited in an opinion text where it is known which factors apply to the cited case can support inferences about the issues and factors of the opinion.*

This hypothesis is fairly specific for the legal domain, and illustrates how one can employ a model of reasoning with cases in this domain. Knowing which factors apply to the cases cited in an opinion is evidence for which issues are dealt with in the citing case, since citations are made to support the court's decision related to an issue by comparing the case to precedents. The information derived from cases cited would not allow us directly to ascertain which factors apply, but it effects our beliefs about the case, as we will illustrate with the *Peggy Lawton* case.

Peggy Lawton cites ten cases, for five of which we have a factor representation. The factors in the cases cited can be related to abstract factors in the Factor Hierarchy. In Fig. 3, for example, the factors in *Dynamics* and *Eastern Marble* are related to F101 via F102 and F104. From the court's selection of cases to cite, we expect it was held that plaintiff's information was a trade secret (F101), that plaintiff took measures to maintain the secrecy of the information (F102) and that the information was valuable (F104).

Further Proposals

We think the task of assigning factors to opinions texts is so complicated and involves so many different aspects that some combination of several or all of the techniques described above is most appropriate, and may help overcome their respective inherent limitations.

Finally, we also attempt to use more detailed information. We are marking up those sentences containing the information whether a factor applies or not. For certain factors, we manually constructed rules similar to Construe (Hayes 1992) to find the relevant sentences, which seems to work fairly well. Relational learning methods (Craven *et al.* 1998) are a promising way to automatically infer the kind of rules we hand-coded. It has to be shown, though, how well they scale up to the very long and complex documents in the legal domain. Alternatively, a case-based approach, as used for machine translation (Brown 1996) may be helpful.

Conclusions

Research on ML for text should not only focus on statistical models for improving upon the algorithms used, but also consider new approaches for integrating background knowledge. ML is a very promising approach for "bootstrapping" a large textual CBR system, by taking a medium sized case-base and induce a classifier to automatically index new texts. However, cases and their representation are often very complex, and the number of training instances in a typical CBR system is too small, so that purely inductive learning algorithms are not sufficient. We presented a set of hypotheses and some initial experimental evidence how adding background knowledge have the potential to be useful for overcoming these problems.

References

Aleven, V. 1997. *Teaching Case-Based Argumentation through a Model and Examples*. Ph.D. Dissertation, University of Pittsburgh, Intelligent Systems Program.

Blum, A. 1995. Empirical Support for Winnow and Weighted-Majority based algorithms. In *Proc. of the 12th International Conference on Machine Learning*.

Brown, R. 1996. Example-Based Machine Translation in the Pangloss System. In *Proccedings of the 16th International Conference on Computational Linguistics*.

Brüninghaus, S., and Ashley, K. 1997. Using Machine Learning for Assigning Indices to Textual Cases. In *Proc. of the 2nd International Conference on CBR*.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 1998. Learning to Extract Symbolic Knowledge from the World Wide Web. To appear in: *Proc. of the 15th National Conference on Artificial Intelligence*.

Hayes, P. 1992. High-Volume Text Processing Using Domain-Specific Techniques. In Jacobs, P., ed., *Intelligent Text-Based Systems*. Hillsdale, NJ: Lawrence Earlbaum.

Joachims, T. 1996. A Probabilistic Analysis of the Rochio Algorithm with TFIDF for Text Categorization. Technical report, Carnegie Mellon University. CMU-CS-96-118.

Koller, D., and Sahami, M. 1997. Hierarchically classifying documents using very few words. In *Proc. of the 14th International Conference on Machine Learning*.

Lewis, D.; Shapire, R.; Callan, J.; and Papka, R. 1996. Training Algorithms for Linear Text Classifiers. In *Proc. of the 19th Annual International ACM SIGIR Conference*.

McCallum, A. 1997. Libbow/rainbow text-classification software package. available from http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html.

Yang, Y. 1997. An Evaluation of Statistical Approaches to Text Categorization. Technical report, Carnegie Mellon University, Pittsburgh, PA. CMU-CS-97-102.