Artif Intell Law (2009) 17:125–165
DOI 10.1007/s10506-009-9077-9

Automatically classifying case texts and predicting outcomes

Kevin D. Ashley · Stefanie Brüninghaus

Published online: 9 July 2009

Abstract Work on a computer program called SMILE + IBP (SMart Index Learner Plus Issue-Based Prediction) bridges case-based reasoning and extracting information from texts. The program addresses a technologically challenging task that is also very relevant from a legal viewpoint: to extract information from textual descriptions of the facts of decided cases and apply that information to predict the outcomes of new cases. The program attempts to automatically classify textual descriptions of the facts of legal problems in terms of Factors, a set of classification concepts that capture stereotypical fact patterns that effect the strength of a legal claim, here trade secret misappropriation. Using these classifications, the program can evaluate and explain predictions about a problem's outcome given a database of previously classified cases. This paper provides an extended example illustrating both functions, prediction by IBP and text classification by SMILE, and reports empirical evaluations of each. While IBP's results are quite strong, and SMILE's much weaker, SMILE + IBP still has some success predicting and explaining the outcomes of case scenarios input as texts. It marks the first time to our knowledge that a program can reason automatically about legal case texts.

Keywords Predicting case outcomes · Classifying case texts · Case-based reasoning

K. D. Ashley (✉)
Learning Research and Development Center, University of Pittsburgh, Pittsburgh, PA, USA
e-mail: Ashley@pitt.edu

S. Brüninghaus
Graduate Program in Intelligent Systems, University of Pittsburgh, Pittsburgh, PA, USA

1 Introduction

Two long-time goals in AI and Law research are classifying case texts automatically and predicting case outcomes in a manner that can be explained in terms attorneys can understand. In pursuing the prediction goal, researchers have enriched the kinds of knowledge of law and legal argumentation that computer programs can bring to bear. Systems can now support reasoning by analogy, presenting alternative reasonable arguments, and generating testable legal predictions. (See, e.g., Ashley 1988, 1990; Branting 1999; Aleven 2003; Chorley and Bench-Capon 2005). They have not been able to do so starting with cases represented in text, however, nor have they been able to integrate the predictions and arguments into plausible explanations that are intelligible to legal practitioners. The research described here describes progress toward realizing both goals.

Integrating text processing with AI and Law models remains a pressing need. Modern text-based legal information retrieval (IR) systems offer a substantial degree of functionality in retrieving relevant case texts given their remarkably limited information about what a text means, and they do so for huge numbers of documents.[1] Although American courts generate a flood of textual case opinions,[2] providers of legal IR process them efficiently, enabling speedy access to new materials within days of receipt. When courts submit opinions in electronic form to the major on-line IR services, the process of adding them to the inverted index is straightforward and requires no textual interpretation; a computer program performs the process automatically (Jackson and Moulinier 2007, pp. 11–12, 26; Turtle 1995, p. 18). Stop words may be removed for indexing (i.e., very frequently occurring words like "and" or "the") and the words may be stemmed to remove endings like a plural—s or past tense—ed. Case opinions are retrieved from the inverted index by matching the terms in queries to those in the index, retrieving the indexed documents, and ranking them according to statistical criteria that capture information about how unique a term is to a particular document.

When it comes to *classifying* the new cases according to some conceptual scheme that summarizes the case's legal significance, the situation is quite different. West's key number system is the paradigm example of a conceptual scheme for legal classification; it makes topical or taxonomic relationships explicit (Hanson 2002, p. 574). Legal researchers can find cases by following relevant key numbers or by submitting a natural language query from which the query terms are extracted, or both. Entering new case texts into a classified index like West's key number system may also be efficient but not necessarily due to automation; humans must still perform the classification manually after reading the texts (Jackson and Moulinier 2007, p. 116). In practice, this manual process does not much delay entering the texts into the legal databases even after they have been automatically entered into

[1] In 1995, U.S. case law comprised about 50 gigabytes of text and was growing by 2 gigabytes per year (Turtle 1995, pp. 6, 47).

[2] "The appellate courts of the United States hand down about 500 decisions a day, so there is considerable pressure upon legal publishers to process case opinions in a timely fashion." (Jackson et al. 2003).

the inverted indexes primarily because the database services hire large staffs of attorney-editors to keep up with the volume (Thompson 2001).

Neither an inverted index nor the classification scheme does much to help legal researchers perform important subsequent steps: comparing the facts of the retrieved case decisions with a problem's facts and drawing analogical inferences from the comparisons about how the problem likely would, or should, be decided. A legal researcher seeks case decisions that are consistent (or inconsistent) with legal propositions she can urge in her current problem and a determination of how strongly the cases support (or oppose) the propositions. Key number topic categories and headnotes summarize some general legal concepts and points of law contained in a case (Thompson 2001), but in order to determine how strongly the case supports the proposition, or its application to the facts at hand, a legal researcher needs to read and compare the case's facts against those of the researcher's current problem.

The work reported here on a computer program called SMILE + IBP (SMart Index Learner Plus Issue-Based Prediction) bridges case-based reasoning and extracting information from texts. From a technological viewpoint, the program addresses a challenging task, but one that is also very relevant from a legal viewpoint: to extract information from textual descriptions of the facts of decided cases and apply that information to predict the outcomes of new cases. The program attempts to *automatically* classify textual descriptions of case facts according to a conceptual scheme of classification concepts, namely, Factors (Ashley 1988, 1990; Aleven 1997, 2003). Factors are often much more fact-specific than the key number abstracts; each one captures a stereotypical pattern of facts that strengthens or weakens a side in a particular kind of legal claim. In this classification scheme, a Factor should not be said to apply unless the stereotypical factual pattern actually is present in the facts of the case (Ashley 1988, 1990). Consequently, if one learns that a set of Factors applies in a problem, one can begin to reason about what issues the Factors raise, what other cases share those Factors with the problem, what the outcomes were in those cases, and whether the same outcome should be assigned to the problem.

This is the kind of reasoning that SMILE + IBP performs. The SMILE part of SMILE + IBP, as indicated in Fig. 1, identifies the Factors that apply given a textual description of the facts of a problem or case. Then, using a conceptual scheme that relates Factors to legal issues, IBP identifies the applicable legal issues in the problem or case, automatically compares the problem and cases on their facts, tests hypotheses about which side should win various issues, and explains those predictions in terms that are legally intuitive. SMILE + IBP is a step in the direction of making more problem-solving legal knowledge available in the automated retrieval, processing, and presentation of relevant cases. Factors are useful classifiers. One may draw analogies between cases in terms of shared Factors and distinguish them in terms of certain unshared Factors (Ashley 1990, p. 175), emphasize or downplay distinctions in terms of legal issues to which the Factors relate (Aleven 1997), justify analogies based on Factors in terms of legal theories or underlying values (Bench-Capon and Sartor 2003; Chorley and Bench-Capon 2005), and predict issue outcomes based on analogous examples while explaining away counterexamples (Ashley and Brüninghaus 2006). As a result, SMILE's

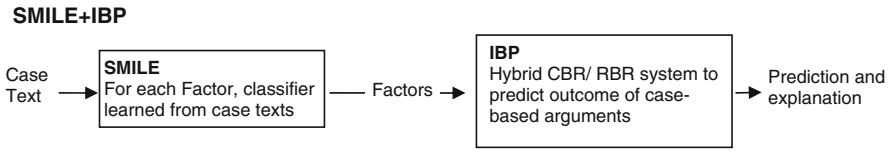

Fig. 1 Overview of SMILE + IBP

classifications of case fact descriptions support analogical legal inference in terms of more abstract legal concepts—and their relationships—than are common in current legal information retrieval methods, thus bridging the gap between case facts and the more abstract legal issues to which they relate.

After summarizing relevant prior work in AI and Law on legal prediction and extraction of legal information from case texts in Sect. 2, there follows a discussion of the SMILE + IBP program starting with a description of the IBP component in Sect. 3. In order to illustrate IBP's hybrid case- and rule-based approach to predicting outcomes of legal cases and explaining the predictions, an extended example, the *National Rejectors* case is introduced. Section 4 describes SMILE's approach to automatically classifying textual descriptions of case facts in terms of Factors. It continues discussing the *National Rejectors* example in order to illustrate the workings of the SMILE program and how, given a textual case as input, it outputs the Factors that apply to the case. In reality, our discussion of SMILE + IBP takes the components out of sequence; given SMILE's output as input, IBP analyzes the case in light of past cases, predicts an outcome, and explains the prediction. Intuitively, however, considering IBP first helps to explain the reason why SMILE was designed and how it functions.

Section 4.2 goes on to explain how SMILE learns the classifiers from a training set of manually-classified sentences and addresses the focal research issue of how best to represent legal case texts for automatic classification and the three alternative text representations we tried, bag-of-words (i.e., a feature list of words), roles-replaced, and propositional patterns. Bag-of-words is an often-used text representation, because it supports a computer program's comparing sentences as alpha-ordered lists of words, that is, as feature vectors (Jackson and Moulinier 2007, p. 30–33). The latter two, invented by us, represent increasingly more knowledge about what the texts mean. Roles-replaced captures information about the roles various actors and objects play in the law suit (e.g., as plaintiff or plaintiff's information); propositional patterns captures information about "who did what" as well as role information. Section 4 ends with a comparison of SMILE + IBP's analysis of the *National Rejectors* case, based on a Factor representation inputted from SMILE, with the IBP analysis of the same case using a Factor representation inputted by a human.

SMILE + IBP has been subjected to extensive evaluation. Section 5 presents an experiment comparing the IBP program's predictions with those of the alternative prediction algorithms introduced in Sect. 2. Then, Sect. 6 presents hypotheses about the relative utility of the three text representations for accurately classifying new textual cases, namely, that our more knowledge-intensive representations

outperform bag-of-words and that propositional patterns outperforms roles-replaced. It describes four experiments designed and carried out to evaluate those hypotheses and reports results in support of our claim that SMILE + IBP analyzes cases input as texts, the first program we know of to do so. As argued in the Conclusions, Sect. 7, SMILE + IBP contributes new techniques for representing and processing cases as texts; these contributions may have importance for attempts to improve computerized legal research.

2 Relation to prior work on prediction and legal information extraction

Prior research in AI and Law has addressed both goals of predicting case outcomes and automatically classifying case texts.

2.1 Prediction

One of the earliest programs in the field of AI and Law employed a nearest neighbor approach to predict outcomes of cases on an issue of tax law using a database of 64 capital gains tax cases represented in terms of 46 descriptors (Ejan Mackaay and Robillard 1974, p. 306). A nearest neighbor approach determines the *k* cases closest to the problem in terms of some similarity measure, and assigns an outcome according to the majority of those cases.[3] This program's similarity metric was a simple count of the descriptors with identical values in the cases. The program generated a visual representation of the case neighborhood suggesting boundaries between pro and con instances where close cases appeared to fall. (Popple 1996, pp. 40–41, 75–82, 87–89, 146–151) applied a nearest neighbor algorithm for prediction using a more complex similarity measure that assigned weights to different fact descriptors. Such weights may correspond to the correlation between features and outcome or the relative spread of the features' values and assign greater weight to closer neighbors or weigh each attribute differently according to some measure of its relevance.

A number of machine learning programs that induce rules or decision trees for predicting case outcomes have been developed. "Machine Learning is the study of computer algorithms that improve automatically through experience" (Mitchell 1997, p. 2). One such program applied ID3 (a predecessor of C4.5 discussed below) to a database of cases involving debt deferral represented in terms of five case descriptors (Zeleznikow and Hunter 1994; Vossos 1995).

Aleven constructed a hierarchical model relating Factors to more abstract representations of their legal significance, the Factor Hierarchy in CATO (Aleven 1997, pp. 44–45; Ashley 2002, p. 181). He used it to predict outcomes of trade secret misappropriation cases (Aleven 2003) according to the following rule: In

[3] A nearest-neighbor classifier has in memory "all the documents in the training set and their associated features. Later, when classifying a new document, *D*, the classifier first selects the *k* documents in the training set that are closest to *D* [using some distance metric like cosine similarity], then picks one or more categories to assign to *D*, based on the categories assigned to the selected *k* documents." (Jackson and Moulinier 2007, p. 144).

order to make a prediction, retrieve the relevant cases according to an argument-based relevance criterion. If there are relevant cases and all had the same outcome, predict that side will win, otherwise abstain. He identified seven argument-based relevance criteria, each of which employed incrementally more knowledge about legal argument (Aleven 2003, 214). For instance, the HYPO-BUC criterion based a prediction on the best untrumped cases (i.e., the cases with: 1. at least one Factor favoring a side, 2. the most inclusive set of Factors shared with the problem, and, in particular, 3. for which there were no cases won by the opposing side that shared with the problem an as-or-more inclusive set of Factors.) The CATO-NoSignDist criterion refined HYPO-BUC, basing predictions on those best untrumped cases that had no significant distinctions from the problem. A distinction was "significant" if and only if it could be emphasized but not downplayed, a criterion defined in terms of the reasons supporting the distinction as represented in CATO's Factor Hierarchy.

With its focus on taking issues into account and testing prediction hypotheses, IBP takes a different approach from the above, versions of all of which were employed in an evaluation of IBP as discussed below. IBP outperformed these approaches; the differences in favor of IBP were significant with respect to all except for one of the rule-learning programs (Brüninghaus and Ashley 2003). In addition, IBP's explanations of its predictions were organized by issues and discussed the hypotheses and possible counter-examples; we argue that its explanations are intuitively more accessible to, and assessable by, attorneys.

Other researchers have implemented prediction based on cases and Factors, but incorporating reasoning about the case in terms of preferences among the underlying normative values at stake (Bench-Capon and Sartor 2001, 2003) and taking into account burdens of proof (Gordon et al. 2007). In modeling legal argumentation as theory construction, that is, in terms of predictive rules induced from the cases, the AGATHA program generated predictions comparable to IBP's in accuracy but for a reduced case base (Chorley and Bench-Capon 2005). Although the program makes use of value preferences, it is not clear it uses values in the same way that attorneys do. The constructed theories do not refer to issues drawn from relevant statutory texts, similar to the Uniform Trade Secret Act or the authoritative Restatement, provisions of which IBP's Domain Model interprets. Thus, it is less clear whether the explanations generated with the induced rules would make sense to attorneys.

2.2 Information extraction from case texts

Information extraction (IE) is a kind of information retrieval that automatically extracts "structured or semi structured information from unstructured machine-readable documents," for instance, recognizing the names of entities such as people, organizations, or products (Jackson and Moulinier 2007, p. 9). Developing a facility for automatically extracting information about Factors from legal cases and for classifying legal cases by Factors would alleviate the bottleneck of manually representing textual cases. It would be an important step toward improving legal information retrieval as well as intelligent tutoring systems for law students.

A considerable amount of AI and Law research has focused on the task of extracting information from legal texts.

The History Assistant extracts from court opinions information about direct history (e.g., "affirmed", "reversed in part", or "motion denied") treatment history (e.g., "overruled", "declined to follow"), and prior cases affected by the direct history (Jackson et al. 2003, p. 241). Somewhat closer to the classification task described in this paper, experiments have shown the feasibility of automatically categorizing legal cases by forty abstract Westlaw® categories (e.g., bankruptcy, finance and banking, criminal justice, insurance), although not yet with commercial accuracy (Thompson 2001, pp. 70–77). More recently, experimenters showed that some linguistic processing, including stemming and part-of-speech tagging helped assign general topic categories (e.g., exceptional services pension, retirement, competence) to legal cases (i.e., decisions of the Portuguese Attorney General's office) (Gonçalves and Quaresma 2005). Both sets of experiments involved automatically assigning abstract topics as categories; in the context of the work reported here, a comparably abstract category would be "trade secret misappropriation". Factors, the categories assigned automatically by SMILE, are considerably more specific, capture more complicated factual patterns, and support a program's reasoning about a case and its outcome.

In order to extract classification information automatically from case texts, one approach involves manually developing rules for automatically identifying patterns of information to be extracted. This kind of template-mining approach was applied in the PRUDENTIA system, designed to add textual cases (Brazilian lower court criminal cases written in Portuguese) to a case-retrieval system (Weber 1998). After interviewing domain experts, the researchers manually constructed pattern extraction rules tailored to the particular type of legal cases. Given a new case text, the program could apply the rules to extract the targeted information automatically. The approach depended on the fact that the case texts were relatively well-structured and tended to use formulaic expressions. Even so, the rules had to be flexible enough to apply to the natural language fillers in the form-like cases; for one feature type, additional processing was required, including parsing. In general, however, the texts were more explicitly structured and simpler than those in the textual summaries of trade secret case facts addressed here, and the patterns to be extracted simpler and more predictable than those associated with Factors.

Where the texts are as complex and lacking in explicit structure as fact descriptions in American legal case opinions, other techniques are required. The SPIRE program employed textual examples of classification concepts and an IR relevance-feedback approach[4] to identify relevant passages in new case texts (Daniels and Rissland 1997). The classifiers were factor-like features that affected the determination of whether a bankrupt debtor had submitted a creditor repayment plan in good faith. Each example was a short textual excerpt relevant to a factor and drawn from cases in the database. SPIRE submitted to the IR system textual

[4] In an IR program with a relevance-feedback module, given feedback from a user in the form of examples of retrieved documents that are relevant to the user's query, the IR system retrieves additional documents similar to the examples.

examples of factors of interest; the system identified similar textual passages in the full text of a new case opinion to be classified. The texts in SPIRE were represented as bags-of-words; in the course of their experimentation, however, the researchers explored some domain-independent variations for improving the basic bag-of-words representation. Significantly, having identified similar factor-related passages, SPIRE could also automatically highlight them in case texts as an aid to human readers. Thus, one could imagine a human legal researcher using SPIRE to help enter a case into a classified index or to read retrieved cases in order to confirm their relevance to an argument.

The work on SMILE + IBP contributes a new approach to representing case texts for the purpose of automated classification; using propositional patterns and role replacement are novel techniques for applying background linguistic and legal knowledge to improve the commonly used bag-of-words representation. It also automates the process of classification more fully; unlike SMILE + IBP, SPIRE could not reason with a case as automatically classified (e.g., it did not predict or explain an outcome for the case).

Other approaches apply machine learning to the tasks of categorizing and extracting information from legal texts, including the use of decision trees and of Naïve Bayes (both of which were used in the SMILE + IBP experiments described below), Support Vector Machines (with kernel functions), context dependent classification and relational learning.

The goal of decision tree learning is to induce decision trees for classifying new instances from the training data (positive and negative textual instances of categorization concepts often represented as feature vectors, that is, alpha-ordered lists of words). A tree-learning algorithm like C4.5 (Quinlan 2004) generates a decision tree whose nodes represent tests (e.g., Is the term "unique" present?). For each possible test outcome, the tree branches into subtrees whose leaf nodes correspond to categorizations (e.g., as an instance of Factor F15, Unique-Product, or not.) The resulting decision tree can easily be converted into if-then rules (Moens 2006, p. 119).

A Naïve Bayes method models the statistical distribution of features (words) within the textual training instances and uses the model to classify new instances. It computes the conditional probability that a textual instance's feature vector belongs to a class, based on the probability of observing feature vectors of textual instances of each class and the prior probability that an instance will be assigned to a class (Jackson and Moulinier 2007, p. 129f).

In a Support Vector Machine, kernel functions model more complex structures in the textual data than are captured in word feature vectors (e.g., parse tree similarity or script trees that capture discourse structures in arguments) and use them to discriminate between positive and negative instances of the categorization concepts (Moens 2006, p. 99ff). The kernel functions represent the positive and negative textual training instances more abstractly in such a way that it is easier to cover all of the positive training instances while not covering any negative ones (Moens 2006).

Context dependent classification methods can also be applied to legal texts represented as feature vectors but where the features and values stored in different

vectors together contribute to the classification using relational models based on statistical techniques, such as hidden Markov models or conditional random fields, to capture some of the inherent domain structure (Moens 2006, p. 107–118, 121f). Both Support Vector Machines and context dependent classification methods tend to work best with larger training sets than we had available.

Other work exploring how to improve legal text categorization by augmenting text representation in different ways includes translating legal texts into graph structures preserving information about word order (Cunningham et al. 2004) and representing legal case texts in a network of concepts and citations (Rose 1994). In work subsequent to SMILE + IBP (Moens et al. 2007) assessed different text representations for classification. The researchers applied naïve Bayes and maximum entropy classifiers to identify argumentative sentences in a corpus of single sentences drawn from court reports and other sources, represented in terms of a variety of features, including: unigrams, bigrams, trigrams, word couples, adverbs, verbs, modal auxillary verbs, text statistics, keywords, parse-tree depth and number of subclauses. A combination of word couples, verbs, and statistics on sentence length, word length, and number of punctuation marks achieved the highest accuracy: 73.75%.

Computational linguists are applying domain independent NLP tools to build systems to automatically summarize legal cases in terms of the most important portions of the opinions (Grover et al. 2003; Hachey and Grover 2006). Their approach attempts to determine the role of a sentence in the legal case, for instance, whether it describes the applicable law or the facts of the case. In this, it is similar to the Salomon program, which extracts from opinions in criminal cases the criminal offenses raised and legal principles applied in order to generate case summaries (Uyttendaele et al. 1998). The information about sentences' roles enables informed decisions about whether to include the sentence in the summary. Most recently, statistical parsing techniques are being applied to extract and interpret sentences describing the holdings of legal cases, a step toward automatically generating a "structured casenote,…a computational version of the traditional 'brief' that first-year law students are taught to write."(McCarty 2007). Unlike these approaches, SMILE + IBP applies a combination of linguistic information and legal background knowledge so that it can reason with the cases it automatically classifies. These other programs cannot reason with the case summaries they produce.

3 Issue-based prediction with IBP

In order to understand SMILE + IBP, it is helpful to begin with the prediction task as performed by IBP. The IBP component integrates case-based reasoning with a logical model of abstract legal issues associated with a legal claim of trade secret misappropriation, hence the name Issue-Based Prediction. The prediction algorithm is shown in Fig. 2. Given a problem case, represented as a set of Factors, IBP identifies the issues raised, and for each one, retrieves cases that share the issue and issue-related Factors with the problem. It then applies a kind of scientific evidential reasoning with the retrieved cases to predict which side should win on that issue.

Input: Current fact situation (*cfs*)
A. Identify issues raised by *cfs* Factors
B. For each issue raised, determine the side favored for that issue using Theory-Testing:
1. if all issue-related Factors favor the same side, then return that side,
2. else retrieve issue-related cases in which all issue-related Factors apply
a. if there are such issue-and-factor-related cases, then form hypothesis that same side *s* will win that won majority of cases
i. if all issue-and-factor-related cases favor side *s*, then return side *s*,
ii. else try to explain away exceptions with outcomes contrary to hypothesis
(a) if all exceptions can be explained away, then return side *s* favored by hypothesis
(b) else, return "abstain"
b. if no issue-and-factor-related cases are found, then call Broaden-Query
i. if query can be broadened, then call Theory-Testing for each subset of issue-related Factors and combine predictions for each set.
ii. else return "abstain"
C. Combine prediction for each issue
Output: Predicted outcome for *cfs* and explanation

Fig. 2 IBP algorithm

Finally, it combines the issue-based predictions to predict the winner of the case and outputs an explanation.

The domain model for trade secret misappropriation, shown in Fig. 3, relates the Factors (and indirectly, the cases to which they apply, that is, the cases indexed and classified by the Factors) to various issues involved in supporting such a claim (Brüninghaus and Ashley 2003). The legal issues and logical relationships in IBP's Domain Model are a distillation and interpretation of two authoritative sources on the law of trade secret misappropriation, a uniform statute[5] and a Restatement provision.[6]

Given information that a new problem involves certain Factors, the Domain Model enables the IBP program to classify it in terms of relevant issues. For example, consider the case of National Rejectors v. Trieman,[7] of which Fig. 4 shows a case squib similar to the kind that first year law students are taught to prepare in briefing cases, including a synopsis of the important facts of the case. Here the synopsis has been annotated to indicate the applicable Factors (the annotation process is discussed in Sect. 4), which are listed and identified in the IBP column of Table 1. IBP's analysis of the case based on these Factors is shown in the left column of Fig. 5. The SMILE + IBP column of Table 1 and the right column of Fig. 5 also show IBP's analysis of a somewhat different version of the *National*

[5] "'Trade secret' means information, […] that: (1) derives independent economic value, […] from not being generally known to, and not being readily ascertainable by proper means […] and (2) is the subject of efforts that are reasonable under the circumstances to maintain its secrecy." Uniform Trade Secrets Act § 1(4) (1985).

[6] "One […] is liable [for trade secret misappropriation if] (a) he discovered the secret by improper means, or (b) his disclosure or use constitutes a breach of confidence […]" Restatement (First) of Torts § 757 (1939).

[7] 409 S.W.2d 1 (Mo. 1966). This case was in the original CATO database. As in CATO, we focus only on the decision of the trade secret misappropriation claim regarding one defendant, Trieman. Other claims are ignored. In CATO, representing inconsistent results of trade secret claims against other defendants would require entering additional versions of the case.

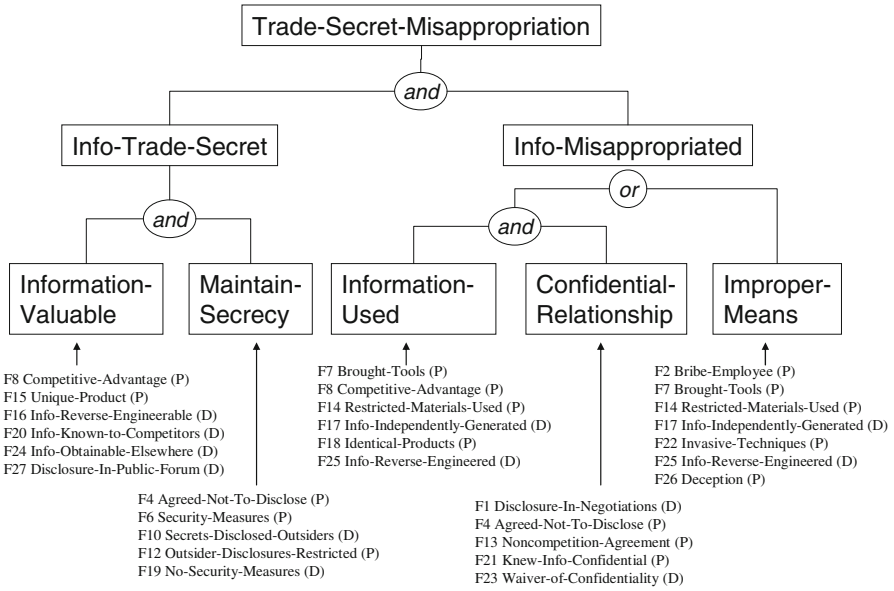

Fig. 3 IBP's Domain model for trade-secret misappropriation

Since the 1940's, National was practically the sole supplier of coin-handling devices, which are used in vending machines, amusement machines, and coin-operated washing machines. [F15] National developed its products (rejectors and changers) through "many years of trial and error, cut and try and experimentation." In 1957, National employees including defendant Trieman, a sales manager, and Melvin, an engineer, started their own business for producing coin-handling devices. … Melvin, working at his home, designed two rejectors that were as close as possible to the comparable National rejectors. [F18] … He also used some National production drawings, as well as a few parts and materials obtained, without consent, from National. [F7] However, none of defendants' drawings was shown to be a copy of a drawing of National. The resulting rejector improved on the National product in certain ways. [Melvin and Trieman resign from National.] National's vice-president testified that the National rejectors could be taken apart simply and the parts measured by a skilled mechanic who could make drawings from which a skilled modelmaker could produce a handmade prototype. [F16] The shapes and forms of the parts, as well as their positions and relationships, were all publicized in National's patents as well as in catalogs and brochures and service and repair manuals distributed to National's customers and the trade generally.[F27] National did not take any steps at its plant to keep secret and confidential the information claimed as trade secrets. [F19] It did not require its personnel to sign agreements not to compete with National. [F19] It did not tell its employees that anything about National's marketed products was regarded as secret or confidential. [F19] Engineering drawings were sent to customers and prospective bidders without limitations on their use. [F10] …

Fig. 4 Annotated *National Rejectors* case squib (Excerpts)

Table 1 Factor representation of *National Rejectors* Case in CATO (by a Human), IBP and SMILE + IBP and as analyzed in IBP and SMILE + IBP

Factors & Issues	Human	IBP	SMILE + IBP
F6 Security-Measures (P)	No	No	√
F7 Brought-Tools (P)	√	√	√
F10 Secrets-Disclosed-Outsiders (D)	√	√	√
F15 Unique-Product (P)	√	√	No
F16 Info-Reverse-Engineerable (D)	√	√	√
F18 Identical-Products (P)	√	√	√
F19 No-Security-Measures (D)	√	√	√
F25 Info-Reverse-Engineered (D)	No	No	√
F27 Disclosure-in-Public Forum (D)	√	√	No
Issue: Info-Used		Factors F7(P) F18(P)	Factors: F7(P) F18(P) F25(D)
		Issue outcome: Plaintiff	Issue outcome: Plaintiff
Issue: Security-Measures		Factors: F10(D) F19(D)	Factors: F6(P) F10(D) F19(D)
		Issue outcome: Defendant	Issue outcome: Abstain
Issue: Info-Valuable		Factors: F15(P) F16(D) F27(D)	Factors: F16(D)
		Issue Outcome: Abstain	Issue outcome: None
Overall case outcome		Defendant	Abstain

Rejectors case that, as described in Sect. 4, the SMILE program generates from its analysis of the squib in Fig. 4 and inputs to IBP. IBP's analysis of both versions of the case, shown in Fig. 5, will be used to illustrate all of the steps of IBP's algorithm.

Step A of the IBP algorithm, Fig. 2, takes the list of *National Rejectors* Factors as shown in cell 1 (or 1′) of Fig. 5 and uses the Domain Model of Fig. 3 to identify the relevant issues in the case. As summarized toward the bottom of Table 1, both versions of the *National Rejectors* case involve the following issues, whether: (1) the alleged trade secret information was used by the defendant, (2) plaintiff took measures to maintain secrecy of the information (also referred to as security-measures), and (3) the information was valuable (i.e., conferred a competitive advantage on plaintiff).

In the hypothesis- or Theory-Testing step, Step B, Fig. 2, the IBP algorithm generates and tests hypotheses predicting which side is favored on each issue. When IBP predicts the outcome of an issue, if all its Factors related to an issue favor one side, plaintiff, say, it would simply predict plaintiff wins that issue (Fig. 2, B.1). For instance, in the *National Rejectors* case, Fig. 5, IBP predicts the outcome of the Info-Used issue for plaintiff (cell 2) and the Security-Measures issues for defendant (cell 3), because for each issue, all of the Factors in the case relevant to that issue favor that side.

IBP Analysis for *National Rejectors* Case as Input by:	
Human	**SMILE**
1. Prediction for NATIONAL-REJECTORS Factors favoring plaintiff: (F18 F15 F7) Factors favoring defendant: (F27 F19 F16 F10)	**1′. Prediction for NATIONAL-REJECTORS** Factors favoring plaintiff: (F18 F7 F6) Factors favoring defendant: (F25 F19 F16 F10)
2. Issue raised in this case is INFO-USED Relevant factors in case: F18(P) F7(P) The issue-related factors all favor the outcome PLAINTIFF.	**2′. Issue raised in this case is INFO-USED** Relevant factors in case: F25(D) F18(P) F7(P) Theory testing did not retrieve any cases, broadening the query. For INFO-USED, the query can be broadened for PLAINTIFF. Each of the pro-P Factors (F7 F18) is dropped for new theory testing. Theory testing with Factors {F7 F25} still does not retrieve any cases. Theory testing with Factors {F18 F25} gets the following cases: (KG PLAINTIFF F6 F14 F15 F16 F18 F21 F25) (MINERAL-DEPOSITS PLAINTIFF F1 F16 F18 F25) In this broadened query, PLAINTIFF is favored. By a-fortiori argument, PLAINTIFF is favored for INFO-USED.
3. Issue raised in this case is SECURITY-MEASURES Relevant factors in case: F19(D) F10(D) The issue-related factors all favor the outcome DEFENDANT.	**3′. Issue raised in this case is SECURITY-MEASURES** Relevant factors in case: F19(D) F10(D) F6(P) Theory testing did not retrieve any cases, broadening the query. For SECURITY-MEASURES, query can be broadened for DEFENDANT. Each of the pro-D Factors (F10 F19) is dropped for new theory testing. Theory testing with Factors {F10 F6} gets the following cases: *[11 cases won by plaintiff, 2 cases won by defendant]* Trying to explain away the exceptions favoring DEFENDANT MBL can be explained away with unshared ko-factor(s) (F20). CMI can be explained away with unshared ko-factor(s) (F27 F20 F17). Therefore, PLAINTIFF is favored for the issue. In this broadened query, PLAINTIFF is favored. Theory testing with Factors {F19 F6} still does not retrieve any cases. There is no resolution for SECURITY-MEASURES, even when broadening the query.
4. Issue raised in this case is INFO-VALUABLE Relevant factors in case: F27(D) F16(D) F15(P) Theory testing did not retrieve any cases, broadening the query. For INFO-VALUABLE, the query can be broadened for DEFENDANT. Each of the pro-D Factors (F16 F27) is dropped for new theory testing. Theory testing with Factors {F16 F15} gets the following cases: [8 cases won by plaintiff] In this broadened query, PLAINTIFF is favored. Theory testing with Factors {F27 F15} gets the following cases: (DYNAMICS DEFENDANT F4 F5 F6 F15 F27) In this broadened query, DEFENDANT is favored. There is no resolution for INFO-VALUABLE, even when broadening the query.	**4′. Issue raised in this case is INFO-VALUABLE** Relevant factors in case: F16(D) The case has only one weak factor related to the issue, which is not sufficient evidence to include this issue in the prediction.
5. Outcome of the issue-based analysis: For issue INFO-VALUABLE, ABSTAIN is favored. For issue SECURITY-MEASURES, DEFENDANT is favored. For issue INFO-USED, PLAINTIFF is favored. => Predicted outcome for NATIONAL-REJECTORS is DEFENDANT	**5′. Outcome of the issue-based analysis:** For issue INFO-USED, PLAINTIFF is favored. For issue SECURITY-MEASURES, ABSTAIN is favored. => Predicted outcome for NATIONAL-REJECTORS is ABSTAIN

Fig. 5 Analyses of *National Rejectors* case (which Defendant won) by: IBP (*left*) versus SMILE + IBP (*right*) given text of *National Rejectors* squib

If an issue in the case has some issue-related Factors that favor plaintiff and some defendant, however, then IBP uses these issue-related Factors to retrieve prior cases involving the same issue and all or some of the same Factors (Fig. 2, B.2.a). This occurs for the Info-Valuable issue (cell 4) and for the Info-Used (cell 2′) and Security-Measures (cell 3′) issues in Fig. 5. In all of these instances, IBP succeeds in finding cases that share some set of issue-related Factors with the problem; call these issue-and-factor-related cases. (The process of trying to find cases that share a

set of issue-related Factors with the problem is called "broadening" and is described more fully below.) For each set of shared issue-related Factors for which IBP finds cases, it proposes a prediction hypothesis in the following way. It counts the issue-and-factor-related cases that favor the plaintiff and those that favor defendant, and proposes as a hypothesis that the side favored by the majority of such cases should win that issue. It then evaluates the hypothesis by focusing on any counterexamples (i.e., the minority of issue-and-factor-related cases that held for the opponent). If there were no counterexamples, the prediction would stand (Fig. 2, B.2.a.i) (i.e., subject to combining with predictions based on the other sets of shared issue-related Factors that result from broadening, described below.) This is what happened in cells 4 and 2′, Fig. 5. In cell 4, for each of the sets of issue-related Factors, {F16 F15} and {F27 F15}, the retrieved cases favored respectively only the plaintiff or only the defendant. In cell 2′ the cases for {F18 F25} both favored plaintiff.

If there are counterexamples, however, IBP uses a kind of analogical comparison to confirm or reject predictive hypotheses based on analyzing the counterexamples. It attempts to distinguish those counterexample cases in a particular way from the problem situation (Fig. 2, B.2.a.ii). In distinguishing counterexamples, IBP looks for alternative explanations of their outcome in order to explain away the exceptions, thus saving its hypothesis. It tries to find strong Factors, (i.e., highly predictive ones) unrelated to the issue IBP is working on, which could independently account for the exception's outcome. For example, in cell 3′, Fig. 5, IBP had to explain away two counterexamples, the *MBL* and *CMI* cases; it does so by identifying "Knock-Out" Factors.

These so-called Knock-Out Factors (KO-Factors) are defined as Factors representing behavior paradigmatically proscribed or encouraged under trade secret law and for which the probability that a side wins when the Factor applies is at least 80% greater than the baseline probability of the side's winning. This probability is computed as the fraction of the number of cases in the collection where the Factor applies and the side won divided by the number of cases in the collection where the Factor applies. The baseline probability is calculated as the number of cases where the side won divided by the number of cases in the collection.[8] If IBP finds a KO-Factor that accounts for the outcome of an exception, it deems the exceptional case to be distinguishable and, thus, not a reason for abandoning its hypothesis. If IBP can distinguish all of the counterexamples, then the hypothesis is treated as confirmed. If it cannot distinguish all exceptions to the hypothesis in this way, it abstains for that issue and set of factors. For example, in cell 3′, Fig. 5, IBP explains away the *MBL* using the KO-Factor F20 and the *CMI* case using the KO-Factors, F17, F20, and F27; the pro-plaintiff prediction is confirmed.

By contrast, there also is a category of Weak Factors for which the probability of the favored side's winning, given that one knows the Factor applies, is less than 20% over the baseline probability of the side's winning. At least for the cases in our collection, the Weak Factors appear to be relevant only in the context of other

[8] IBP's list of KO-Factors includes: F8 Competitive-Advantage (P) (i.e., defendant saved development time and money by using plaintiff's information), F17 Info-Independently Generated (D), F19 No-Security-Measures (D), F20 Info-Known-to-Competitors (D), F26 Deception (P), F27 Disclosure-In-Public-Forum (D).

Factors; courts appear not to have treated them as sufficient on their own to raise an issue. This has led to a policy: if the only Factor concerning an issue is a Weak Factor, IBP will not regard the issue as having been raised and will not propose a hypothesis with respect to that issue. For example, if a case has Factor F10 Info-Disclosed-Outsiders (D) but no other Factors related to the Security-Measures issue, IBP does not propose a hypothesis for that issue. This is illustrated in cell 4′, Fig. 5.[9]

In Step B.2 of Fig. 2, it often occurs that no case in the database shares *all* of the issue-related Factors in the problem. In that case, IBP attempts to broaden the query by relaxing its constraints (Fig. 2, Step B.2.b) in order to determine whether it can test a more general but still pertinent hypothesis. For instance, if two or more issue-related Factors favor a side, IBP removes each one in turn from its query. If it finds cases responsive to the broader query (i.e., that share the less inclusive set of Factors), it tests the hypothesis as above. In each of cells 4, 2′, and 3′ of Fig. 5, IBP had to broaden the query to find relevant cases.

Broadening a query makes sense; if a case retrieved by a broader query favors that side, one may conclude the problem is even stronger for that side given the dropped Factors. This is illustrated in cell 2′, Fig. 5. If the result of broadening a query for one side is that the other side is favored, as in cell 3′, Fig. 5, the issue is unresolved and IBP abstains. As a result of broadening, there may be multiple hypotheses per issue that need to be combined. If a query can be broadened for both sides, as in cell 4, Fig. 5, IBP abstains on the issue (Fig. 2, Step B.2.b.ii).

In Step C, Fig. 2, IBP combines the hypotheses for all of the relevant issues according to the logical connections in the model (Fig. 3) and makes an overall prediction, which it explains by recounting the above predictions, as in the *National Rejectors* analysis, Fig. 5.[10] In cell 5, since IBP predicted the Security-Measures issue would go for defendant, and since plaintiff must win that issue to win the claim (Fig. 3), IBP predicts defendant wins the claim, a prediction that turns out to be correct. Given the SMILE version of the *National Rejectors* case, however, IBP abstains on the Security-Measures; thus it does not have enough information to predict an outcome on the claim and it abstains.

4 The SMILE program

SMILE addresses the challenge of processing the textual descriptions of case facts and identifying the patterns of facts associated with known Factors. Its role is to automatically classify legal case texts by the Factors that apply. As Fig. 1 indicates, the program processes the text of a case squib (like the *National Rejectors* squib in Fig. 4), determines the applicable Factors, and inputs them to IBP. The work on SMILE has focused on the threshold issue of how best to represent the texts in order to support automated classification. IBP's prediction function is used as one among

[9] The other Weak Factors include F1 Disclosure-in-Negotiations (D), and F16 Info-Reverse-Engineerable (D).

[10] For other examples of the kind of explanations IBP generates to justify its predictions, see (Ashley and Brüninghaus 2006, p. 331).

a number of metrics for objectively evaluating how well, as compared to human-assigned case classifications, SMILE performs its text classification function using various plausible text representation techniques, and for determining the best of the three text representations.

In understanding SMILE's approach, it is useful to recall the limitations of computational natural language processing (NLP), that is, systems that map human language into internal representations that computer programs can manipulate appropriately to perform a task (Dale 2000, v–viii). The complexity of legal opinions and of the real world situations they describe make them extreme examples of the problems of syntactic, lexical, and semantic ambiguity that plague attempts at computational natural language processing (Jackson and Moulinier 2007, pp. 3–4; Dale 2000, p. 11). A system needs to choose among: multiple possible grammatical parses of a sentence, the words' multiple dictionary meanings, the possible referents of pronouns, and the possible meanings of the sentence. For even moderately simple texts, the combinations of these choices explode rapidly, potentially swamping the computer's ability to keep track of them (Dale 2000, v–viii).

While these technical problems are increasingly being addressed through parallel processing and statistical NLP techniques (see supra, Sect. 2), in the absence of robust natural language processing for legal texts, we used a combination of shallow parsing, information extraction, and machine learning techniques. A shallow parser identifies grammatical constituents in sentences (e.g., noun groups, verbs, prepositional phrases) but does not necessarily specify their internal structure or roles in the sentence (Jackson and Moulinier 2007, p. 16).

As illustrated at the top of the SMILE system overview, Fig. 6 (and described more fully below), given a training set of cases whose sentences have been manually marked up as positive or negative instances of Factors, SMILE uses shallow parsing to represent the sentence texts in terms of certain patterns. The

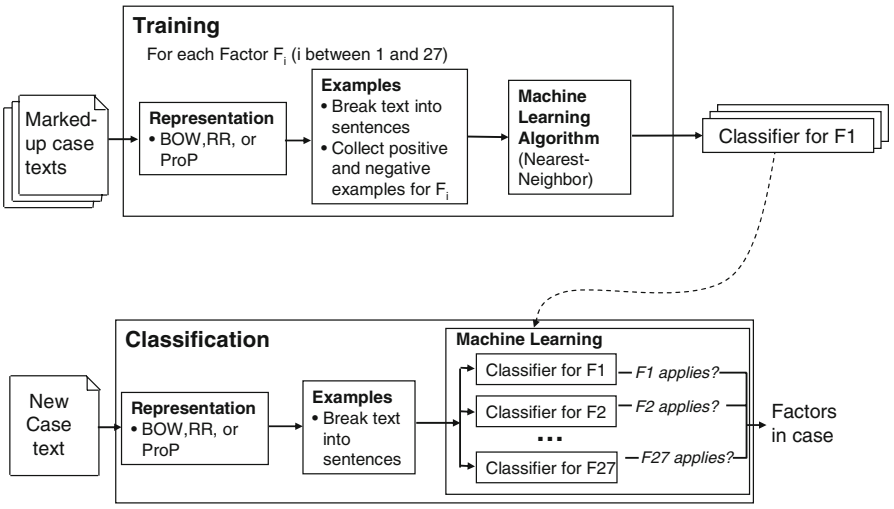

Fig. 6 Overview of SMILE system

instances are generalized using information extraction to substitute party names and product information with information about their roles in the cases. The program then applies a machine learning algorithm to help it match similarly represented sentences in the texts of new cases to those in the training set. In this way, it automatically classifies cases by the Factors disclosed in the text; it applies the Factors whose instances are the best matching sentences to those in the new case.

Here is an example of SMILE's task. If the defendant in a trade secret misappropriation case signed a nondisclosure agreement, that is an important, if not decisive, strength in favor of the plaintiff. The Factor F4, Nondisclosure-Agreement (p), is intended to capture a fact pattern in which, "Defendant entered into a nondisclosure agreement with plaintiff" (Aleven 1997, Appendix 2). F4 is not intended to apply where the plaintiff "obtained nondisclosure agreements from other employees but not from the defendant"; that is, the Factor does not apply when a group of individuals had signed nondisclosure agreements, but defendant was not explicitly mentioned. Obtaining nondisclosure agreements from other employees also helps plaintiffs, but it is covered by another Factor, F6 Security-Measures (p). Some positive examples of F4 include:

1. Newlin and Vafa had signed nondisclosure agreements prohibiting them from using ICM software and tools upon leaving ICM.[11]
2. Ungar signed a nondisclosure agreement.[12]
3. […] defendant Hirsch was employed by plaintiff, where he executed a nondisclosure agreement; […][13]

The negative examples of F4 include all the sentences of the cases in the training set that have nothing to do with nondisclosure agreements, as well as sentences that refer to nondisclosure agreements but do not indicate they were entered into by the defendants, such as:

4. Surgidev requires its employees to sign nondisclosure agreements, […][14]
5. Employees were required to sign nondisclosure agreements.[15]

SMILE's task is to learn from these positive and negative examples of Factor F4 a classifier that can be applied to new case texts to determine if F4 applies in them.

4.1 Overview of SMILE

As noted above, SMILE learns to classify new case texts from a training set of manually-classified case texts. The training set comprises 146 case texts that were

[11] Integrated Cash Management Services, Inc. v. Digital Transactions Inc., 732 F.Supp. 370 (S.D.N.Y. 1989).

[12] Henry Hope X-Ray Products v. Marron Carrel, Inc., 674 F.2d 1336, 1339 (9th Cir. 1982).

[13] E. I. duPont de Nemours v. American Potash & Chemical Corp., 200 A.2d 428, 535 (Del. Ch. 1964).

[14] Surgidev Corp. v. Eye Technology, Inc., 648 F.Supp. 661, 675 (D.Minn. 1986). In this case, the opinion indicates that the employee individual defendants *did* sign nondisclosure agreements, but the squib-writer did not indicate that the employees were also defendants.

[15] USM Corp. v. Marson Fastener Corp., 379 Mass. 90, 393 N.E.2d 895, 899 (1979).

prepared and classified for use in the CATO intelligent tutoring system that taught law students basic argumentation skills using trade secret misappropriation cases (Aleven 1997, Appendix 4). As shown at the top of Fig. 6, for each of the twenty-six Factors, a separate classifier is constructed based on the set of positive and negative training instances for that Factor. In each training case, all of the sentences from which it could be reasonably inferred that a Factor applied in the case were manually marked-up as positive instances of that Factor. All the rest of the sentences were treated as negative instances of the Factor.

As noted, the case texts are squibs like those that first year law students prepare in briefing cases, including a synopsis of a case's important facts.[16] Originally, the squibs were employed as teaching aides in the CATO system. Law students hired specially for the task were instructed to prepare narrative summaries of each case's important facts. Apprised of the twenty-six Factors concerning trade secret law, the student squib-writers were encouraged to find and include fact descriptions in the opinions related to any applicable Factors, but they were also instructed to include all facts that seemed important to the decision whether or not there were corresponding Factors. The students were encouraged to cut-and-paste relevant short passages from the case opinions but to incorporate them seamlessly into a readable narrative. The use of squibs in the SMILE experiments and not full-text case opinions limits the generality of the results; given the limitations of state-of-the-art technology for natural language processing and information extraction from text, the use of squibs was a good place to start.

For the SMILE project, student coders marked up all of the squibs to indicate positive instances of Factors. For instance, Fig. 4 above has already shown the annotated squib for the *National Rejectors* case. The mark-up indicates the sentences the student coders identified as justifying the conclusion that seven Factors apply in the case (shown in the Human column of Table 1), three of which happen to favor plaintiff and four the defendant.[17]

When the SMILE program classifies a new case text, as suggested in Fig. 6 at the bottom, the classifiers it has learned for all twenty-six Factors are applied to each of the new case's sentences. The program employs a nearest-neighbor algorithm (implemented in a program called Timbl (Daelemans et al. 2004; 2007)) to determine which Factors apply.[18] Since a number of the Factors have very few positive instances, a nearest-neighbor approach seemed appropriate; it had been applied with good results for an information extraction task where negative

[16] For a description of CATO and additional examples of its squibs, see (Ashley 2000, 284–288; Ashley and Brüninghaus 2006).

[17] In Table 1, the representation of the *National Rejectors* case as a training instance for the SMILE program differs slightly from the original CATO version, shown in the human column, input to the IBP program by the addition of F6 Security-Measures (D). There is an inconsistency between F6 Security-Measures (P) and F19 No-Security-Measures; courts sometimes focus in the same opinion on the security measures plaintiff took, justifying application of F6, and the sometimes many security measures that could have been taken but were not, justifying F19. In IBP, the inconsistency was resolved in favor of whichever the court seemed to emphasize more; for SMILE, the inconsistency was allowed to stand.

[18] In generating the Factor classifiers, two other machine learning algorithms were tried, C4.5 (Quinlan 2004), and Naïve Bayes as implemented in Rainbow (McCallum 2004). The nearest-neighbor algorithm worked best.

examples of classification concepts far outnumbered positive examples and the goal was to find the minority class (Cardie and Howe 1997).

For a given Factor classifier, as described more fully below, the algorithm compares the new sentence to each of the positive and negative instances to find the particular instance that was nearest (i.e., most similar) to the new sentence. Then, it classifies that sentence as an instance of the Factor or not based on the classification of the nearest instance. The program classifies the new case text under all Factors for which it contained at least one sentence that was a positive instance of the Factor.

4.2 Representation of case texts

As noted, the goal of SMILE is to assign Factors automatically by applying classifiers to the textual description of the case facts, classifiers it has learned from prior manually-classified case texts like that illustrated in Fig. 4. In an automated approach that learns text classifiers, much depends on (1) how the sentences are represented for purposes of the learning algorithm, here for comparison using the nearest neighbor algorithm, (2) what features of the sentences are compared, and (3) what "most similar" means in assessing the comparisons (see, e.g., Jackson and Moulinier 2007, pp. 31–33, 128, 143–146). As it happens, the most useful and practical representation of sentences for purposes of classification is still an open research question in the field of text-based information retrieval.[19]

We experimented with three plausible representation schemes, representing each sentence:

(1) as a "bag of words" (BOW),
(2) with "roles replaced" (RR), or
(3) as "propositional patterns" with roles replaced (ProPs).

Each representation is defined below or illustrated with a simplified example sentence drawn from MBL (USA) Corp. v. Diekman,[20] (cited in Fig. 5): "Diekman signed a nondisclosure agreement," a sentence that would be classified as an instance of Factor F4 Nondisclosure-Agreement, a Factor that favors plaintiff (P).

A bag-of-words is the simplest representation; as noted, it treats the words of the sentence, irrespective of their order, as the key representational features (Jackson and Moulinier 2007, p. 126). In generating the BOW representation for our experiments, punctuation, numbers, and duplicate words were removed, but we did not carry out stemming or remove stop words. As a result, the example above is represented: "`a agreement Diekman nondisclosure signed`".

In the roles-replaced representation,[21] like BOW, the features representing the sentence are again its words irrespective of order, but there is an important

[19] The most commonly used text representation techniques in information retrieval include feature lists of words (i.e., bag-of-words), syntactic structure, frames/scripts, logic propositions, and network structures (Turtle 1995 p. 18).

[20] 445 N.E.2d 418 (Ill. App. 1 Dist. 1983).

[21] The roles-replaced and propositional patterns text representations were introduced in (Brüninghaus and Ashley 2001, 47–48).

difference: case-specific names of parties and their products have been replaced with their roles in the case, for example, "`a agreement defendant nondisclosure signed`". For purposes of the SMILE experiments, in generating the RR representation, we manually replaced the party and product names in the texts.[22]

We expected that replacing names by roles would help generalize from examples, detect patterns that would otherwise remain hidden, and prevent spurious generalizations based on coincidental occurrences of the same name in different cases. For example, substituting roles for names in the five examples of Factor F4 Nondisclosure-Agreement (p) at the beginning of Sect. 4, makes the pattern, and the distinction between the first three positive examples and the last two negative examples much clearer:

1. Defendant and defendant had signed nondisclosure agreements prohibiting them from using plaintiff information upon leaving plaintiff.
2. Defendant signed a nondisclosure agreement.
3. Defendant was employed by plaintiff, where he executed a nondisclosure agreement;
4. Plaintiff requires its employees to sign nondisclosure agreements.
5. Employees were required to sign nondisclosure agreements.

While the above examples are perfectly intelligible to humans, it is still not clear to the computer who did what to whom. That is the function of the propositional patterns representation. Instead of words, the features representing the sentence are "propositional patterns" or ProPs, combinations of words that fall within four predefined syntactic relationships: subject—verb, verb—object, verb—prepositional phrase, and verb—adjective. The basic ProP representation of the example sentence is "(`defendant sign`) (`sign nondisclosure_agreement`)".

Generating ProPs is more complicated than generating the BOW or RR text representations and involves some natural language processing. As an example of how ProPs are made, consider another example sentence, "The product did not have unique features."

First, a parser (i.e., Sundance (Riloff and Phillips 2004)) generates a shallow parse of the sentence, identifying: (1) the syntactic constituents of the sentence such as, subject, active verb or passive verb, direct object, and (2) providing some information about each, such as the head word that serves that role in the sentence, the root form of the head word (e.g., infinitive form of the verb), and modifying articles, adjectives and adverbs. For the sample sentence, the shallow parse identifies "product" as the subject, "did have" as the active verb, "not" as an

[22] We developed automated information extraction techniques that use heuristics to perform the role substitution task for parties' names and products (Brüninghaus and Ashley 2001). For the experiments reported here, however, we carried out all of the role replacements manually in order to make sure that the texts were as accurate as possible. Since the goal of our experiments here was to test whether role replacement improved Factor assignment, we believed that it was important to eliminate automated role replacements as a source of errors. If SMILE were to be used in larger-scaled applications, automatic role replacements would be necessary. Other commercial and research approaches exist for the task of performing the role replacements, which is a subset of the long-studied problems of named entity recognition and coreference (Jackson and Moulinier 2007, pp. 170–183).

adverb modifying the verb, "features" as a direct object and "unique" as an adjective modifying the direct object.[23]

Second, SMILE analyses the shallow parse to identify instances of particular patterns, including subject—verb, verb—object, verb—prepositional phrase, and verb—adjective. For each pattern it finds, it creates a ProP, appending the head words together in the typical default order of the constituents in a sentence (i.e., subject—verb—object—prepositional phrase). This yields the following ProPs: `(product have)` `(have feature)` `negation` `unique`.

Adding "`negation`" indicates the presence of "not" in the verb—object pattern. If a clause contains a word that the dictionary indicates is a negation, SMILE adds the negation ProP to indicate that something has been negated in the clause. This is a rough heuristic for including negation in the ProPs. Negation is important in assigning Factors. For instance, from the viewpoint of trade secret law, it matters that the product does *not* have unique features. Determining the scope of negation in a sentence, however, is a hard computational problem (Chapman et al. 2001). It is an empirical question whether this heuristic method for approximating the effect of negation is adequate.

Adding modifiers like "`unique`" reflects another rough heuristic. Certain adjectives have a kind of formulaic significance for some of the Factors: unique, confidential, similar, and secret. Since the ProP employs only head words, it cannot reflect the fact that "unique" modifies features in the sample sentence. To compensate for that, when the special adjectives are identified in the sentence, it adds a separate ProP to indicate that one of the clauses contained this modifier.

Finally, a number of objects and actors in trade secret law cases may be described with any of a small set of synonyms or more general terms. For instance, a nondisclosure agreement is a kind of agreement or contract, which may also be referred to as a pact, deal, covenant, compact, settlement or arrangement according to a legal thesaurus. Since sentences in different cases may refer to agreements in all of these ways, we need to develop a technique for broadening the possible matches.

Adding semantic information from an online thesaurus may help a text retrieval program deal with mismatches between a query's terms and those of relevant documents. For instance, it may expand the query to include synonyms of the query's terms. Although the approach sometimes causes more retrieval problems than it solves (Jackson and Moulinier 2007, p. 53), some research has shown that adding semantic information from WordNet improved performance in textual case retrieval (Burke et al. 1997; Lenz 1999). In principle, it could also help a classification program find commonalities between examples that use a different vocabulary. In past experiments, we found that incorporating information from a legal thesaurus led to performance improvements in classification by Factors

[23] In the parsing process, some identification of common phrases needs to take place. For instance, common constituents of trade secret cases described with multiword phrases such as "nondisclosure agreement" need to be recognized as individual entities (i.e., as a single head word like "nondisclosure_agreement") for purposes of parsing and constructing ProPs. So far, we have handled that on an *ad hoc* basis, in effect manually substituting "nondisclosure_agreement" when a parse identifies "nondisclosure" as an adjective modifying agreement. Automating the common phrase recognition task should be possible, but we have not done it. See, (Kim and Wilbur 2000).

(Brüninghaus and Ashley 1999). To accommodate this, when SMILE generates a ProP using a term for which it has a list of synonyms and more abstract terms, it also generates ProPs substituting each of those terms, as well. Thus, to the ProPs for representing the sentence, "Diekman signed a nondisclosure agreement," SMILE adds ProPs for the terms related to agreements.

SMILE also has techniques for generating ProPs for passive verb forms, sentences with multiple clauses, and sentences with multiple verbs. Basically, it breaks complex sentences into clauses and adds ProPs for the clauses, but the details are omitted here.

4.3 Comparing SMILE's National Rejectors analysis with IBP's

While its automatic classification of case texts is very imperfect, this does not necessarily prevent SMILE + IBP from capturing a gestalt-like essence of a case, correctly identifying its issues, if not its outcome, and providing a reasonable explanation. Take for example the *National Rejectors* case. Table 1 (right column) shows the Factor representation of *National Rejectors* as assigned by SMILE and input to IBP as well as SMILE + IBP's predictions per issue (bottom right). SMILE's automatically assigned Factor representation is somewhat different from the CATO version manually assigned and input to IBP ("IBP" Column). SMILE misses F15 Unique-Product (P) and F27 Disclosure-in-Public-Forum (D) and it added F6 Security-Measures (P) (see *supra*, note 17) and F25 Info-Reverse-Engineered.

According to (the unofficial) abstract of the case that appears with the published version, "evidence established that there were no actual trade secrets with respect to plaintiff's [products] and that, although individual defendants, former employees of plaintiff, had improperly used plaintiff's materials and drawings in production of products to compete with plaintiff's products, where plaintiff had not considered information regarding its products to be trade secrets, no warning had been given against use of information." In the terminology of IBP's domain model, the corresponding issues include whether security measures had been taken (Security-Measures), whether the information was valuable (Info-Valuable), and whether defendants had used the information (Info-Used).

As shown in Fig. 5 and summarized in Table 1, IBP's analysis of the manually represented case finds these issues. It reasons that plaintiff is favored for Info-Used, but that the defendant is favored for the issue Security-Measures. With respect to Info-Valuable, which has Factors for both sides, IBP cannot conclude which side is favored and abstains. Overall, IBP's analysis corresponds to the court's opinion and leads to a correct prediction in favor of defendant Trieman on the trade secret misappropriation claim.

SMILE + IBP's automated analysis identifies the same issues. On the other hand, the Factors it misses or adds lead it to abstain from a prediction. Using SMILE's version of *National Rejectors*, IBP correctly identifies the issue Info-Used, and correctly predicts that plaintiff was favored. On the issue of Info-Valuable, however, IBP only sees the weak Factor F16, Info-Reverse-Engineered (d), and does not include the issue in its prediction. Although IBP succeeds in finding the

issue Security-Measures, the presence of the incorrectly assigned Factor F6 complicates the analysis, prevents IBP from resolving the conflicting Factors, and it abstains. As a result, SMILE + IBP abstains from making an overall prediction for *National Rejectors*.

In analyzing the Security-Measures issue, SMILE + IBP trips over the inconsistency between CATO's representation and SMILE's mark-up conventions. As noted, cases may have textual evidence for F6 as well as for F19. SMILE can assign both Factors to a case but for CATO (and thus for IBP alone) F6 and F19 are mutually exclusive. As a practical matter, it is difficult to implement a principled strategy for SMILE's choosing between F6 and F19 without deeper reasoning and an even more informative knowledge representation. In order to maintain IBP's accuracy and reliability, we did not attempt to resolve the conflict heuristically as a result of which the program abstains on the issue.

In considering our argument that SMILE + IBP automatically assigns Factor classifiers that are more specific than those assigned by human indexers using the West key number system, it is interesting to peruse the key number classifications and abstracts under which the *National Rejectors* case is indexed. The West indexing does a good job of identifying the general rules and issues of trade secret law that apply to the case (See e.g., 29Tk413 k. What Are Trade Secrets or Other Protected Proprietary Information, in General; 29Tk419 k. Vigilance in Protecting Secret; Abandonment or Waiver; 231Hk306 k. What Are Trade Secrets or Confidential Information of Employer; 212k56 k. Disclosure or Use of Trade Secrets.) Some of the abstracts provided with the key number classifications provide specific facts from which one could infer how the issues were decided in the case. However, the classifiers do not capture the impact of those facts in a manner that would support prediction. For instance, one factually informative abstract is associated with an abstract classifier about the "Weight and Sufficiency of Evidence".[24] Another is associated with a more specific classifier called "Disclosure or Use of Trade Secrets", but the classifier merely identifies that the topic applies; it does not imply that there was or was not such a disclosure or use.[25] Thus, information potentially useful for prediction is "locked up" in text and cannot be used to guide automated retrieval and inference without techniques like those pioneered in SMILE + IBP. On the other hand, the West key numbers and key

[24] "⊂29T Antitrust and Trade Regulation, ⊂29TIV Trade Secrets and Proprietary Information, ⊂29TIV(B) Actions, ⊂29Tk429 Evidence, ⊂29Tk432 k. Weight and Sufficiency of Evidence. … Evidence established that there were no actual trade secrets with respect to slug rejectors and electrical coin changers, in action based upon alleged misappropriation of trade secrets.".

[25] "⊂212 Injunction …, ⊂212II Subjects of Protection and Relief, ⊂212II(B) Matters Relating to Property, ⊂212k56 k. Disclosure or Use of Trade Secrets. … Although individual defendants, former employees of plaintiff, had improperly used plaintiff's materials and drawings in production of products to compete with plaintiff's products, where plaintiff had not considered information regarding its products to be trade secrets, no warning had been given against use of information, and key patent relating to plaintiff's devices had expired, plaintiff was not entitled to injunctive relief but was entitled to damages based upon profits which it lost by reason of competition of defendant corporation formed by individual defendants, during time that defendant corporation would not otherwise have been in production but for assistance obtained from use of plaintiff's drawings and materials."

abstracts focus on all of the claims, parties, and possibly inconsistent decisions in the case, complicating the indexing task.

5 Evaluating IBP

We undertook two sets of experiments, one to evaluate IBP's predictions as compared to other approaches to legal prediction, and the other to evaluate how well SMILE learned to classify Factors in case texts. IBP's predictions and explanations are interesting in their own right, and they also provide a metric for assessing the adequacy of SMILE's classifications.

The goal was to evaluate the assumption of IBP that testing predictive legal hypotheses against data and attempting to explain away counterexamples is an effective way to implement prediction. We compared IBP's predictions to those of various computerized techniques for predicting case outcomes from a database of labeled examples. The algorithms have complementary strengths and make somewhat different use of the information in the case database. Specifically, we compared IBP to the algorithms in Table 2.

Three of these algorithms induced rules or classifiers that can be expressed as rules: C4.5, a decision-tree learning algorithm that tends to overfit the data by inducing overly specific trees (Quinlan 1993); Ripper, designed to incrementally add relatively simple rules that cover the training data in a patchwork-like manner (Cohen 1995a, b); and RL (Provost et al. 1999); RL derives candidate rules from the labeled examples in the training set, keeps track of their success rates, and assigns weights to rules to help deal with conflicting evidence, all strategies that may be appropriate in complex domains like law where multiple aspects of a case may contribute to its outcome.

Table 2 Prediction algorithms compared to IBP

Algorithm	Type	Description
C4.5	Machine learning	Induces decision trees
Ripper	Machine learning	Induces rules
RL	Machine learning	Induces rules
NN/IB1	Case-based	Nearest neighbor
HYPO-BUC	Case-based	Arguments based on best on-point cases
CATO	Case-based	Arguments based on best on-point cases without significant distinctions
Logistic Regression	Statistical	Statistical using logistic regression
IBP	Case-based	Issue-based hypotheses tested against cases
IBP-cases	Case-based	Hypotheses without issues tested against cases
IBP-model	Case-based	Issue-based hypotheses not tested against cases
Naïve bayes	Machine learning/statistical	Uses statistical weighting
Baseline	Other	Predict majority class (plaintiff)

Three of the algorithms were case-based learning algorithms: IB1 implements a nearest neighbor approach. The similarity measure is the Euclidean (or straight-line) distance between the examples as represented as the endpoints of vectors in a space with as many dimensions as Factors. As explained in Sect. 2, the HYPO-BUC and CATO-NoSignDist algorithms do not compute numerical similarity but rely instead on symbolic reasoning and arguments comparing cases in terms of Factors. The former bases its predictions on the best on-point cases as defined in (Ashley 1988, 1990); the latter focuses on the best on-point cases that are not distinguishable (Aleven 1997).

Two of the algorithms can be characterized as statistical learning methods: Logistical Regression employed a data mining package to perform logistic regression (Witten and Eibe 2005). It was run using a standard automated method without performing any manual trial-and-error process of feature selection. Naïve Bayes, the statistical machine learning algorithm, calculates the probability of each side's winning, using statistical inference, in particular Bayes Rule. (Mitchell 1997). Assuming independence among the Factor descriptors, probabilities for individual Factors are computed simply by counting cases in the database.

We also included two variations of the IBP algorithm, IBP-Case and IBP-Model, discussed below. Finally, as a baseline, we employed an algorithm that simply predicted the majority class regardless of the facts of the new problem. In our database, that means always predicting that plaintiff wins. For a more detailed comparison of the algorithms, see (Ashley and Brüninghaus 2006).

5.1 Experimental design

Each algorithm ran using the same experimental setup on the same database of cases, CATO's database comprising 184 trade secret misappropriation cases of which 108 cases were won by plaintiff, 76 by defendant. The case database was assembled before the invention of IBP; 148 cases were assembled for the original CATO program and 36 additional cases were added later for a different purpose. The database includes cases from a variety of state and federal courts and at a variety of levels from trial courts through the highest appellate courts; for any given case, only the opinion on substantive issues of the highest level court to consider the case is included in the database. The cases range mainly from the 1970s through 1990s, although a small number are even earlier. For purposes of these experiments, no account has been taken of the cases' dates, jurisdiction, or court level. Each case was labeled with its outcome, plaintiff of defendant, and represented as an ordered list of binary features.

As per usual in machine learning experiments, we ran the experiments in a leave-one-out cross-validation, thus guaranteeing that the training set and test set are disjoint (Cohen 1995a, b). A cross-validation design affords maximal use of the available data—every case in the database is used as an input case and as a training case—while ensuring that the program uses no information previously derived from a given case when analyzing that case as an input case. For each of the above algorithms, we repeated the following steps once for each case (i.e., 184 times):
1. The case was designated as the test case and taken out of the database hiding its

outcome. 2. The remaining 183 cases were used as a training set for the algorithm. 3. The trained algorithm predicted the outcome of the test case. 4. The predicted outcome was compared to the test case's previously hidden real outcome to see if the prediction was correct, a mistake or an abstention and the result was recorded. 5. The test case was then returned to the database.

5.2 Results

For each algorithm, the number of abstentions, correct and erroneous predictions and the resulting accuracy are reported in Table 3 and graphed in Fig. 7. Accuracy is defined as the ratio of the number of correct predictions divided by the sum of the correct predictions, incorrect predictions and abstentions (i.e., 184). IBP's accuracy was 91.8%; it predicted the outcome of 169 cases correctly, made 14 errors and abstained once. Second-place RL's accuracy was 88%, followed by Naïve Bayes with 86.4% accuracy. Of the other rule learning programs, C4.5 did better, coming in fourth with an accuracy of 85%; Ripper achieved 83%.

In order to compute whether the differences in the algorithms' performance were statistically significant, we employed McNemar's test (Dietrich 1996). For each incorrectly predicted case, we scored which algorithms did better. Since some of the algorithms abstained, we counted a correct prediction as "better" than an abstention and an abstention as "better" than an error. As reported in Table 3, the difference between the accuracy of IBP and RL, the runner-up, was only marginally significant with $p = 0.08$. (Generally, $p < 0.05$ is considered convincing evidence that there is a true difference between the algorithms.) The other differences were significant. (The software we used did not allow for a comparison between IBP and Logistic Regression.) Thus, except for IBP versus RL, one may conclude that the observed differences are caused by systematic differences between the algorithms rather than natural variance in the data.

Table 3 Correct predictions, abstentions, errors, accuracy and significance for each prediction algorithm

Algorithm	Correct	Abstain	Errors	Accuracy	Significance
IBP	169	1	14	0.918	–
RL	162	0	22	0.880	0.08
Naïve bayes	159	0	25	0.864	0.03
IBP-cases	144	30	10	0.783	0.00
CATO-NoSignDist	143	22	19	0.777	0.00
C4.5	156	0	28	0.848	0.01
Logistic regression	154	0	30	0.837	n/a
Ripper	152	0	32	0.826	0.00
Nearest neighbor	151	0	33	0.821	0.00
HYPO-BUC	125	50	9	0.679	0.00
IBP-model	132	38	14	0.717	0.00
Baseline	106	0	78	0.576	0.00

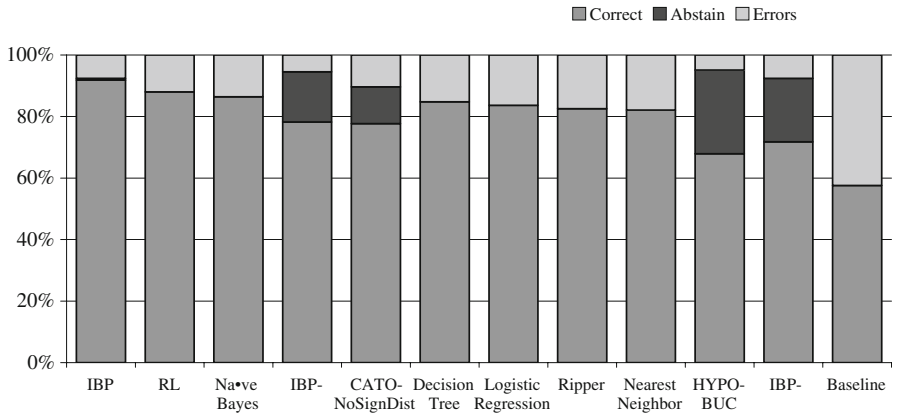

Fig. 7 Comparison of prediction algorithm accuracy

5.3 Discussion

We ruled out the possibility that IBP had been optimized with respect to these cases. As noted, 36 cases had been collected for a different purpose and were not added to the collection until after IBP was completed. We found that IBP's predictions for these 36 cases were no less accurate than for the initial 148.

As noted, two other versions of IBP were tested. In IBP-Cases, the predictions were made with cases alone; all information about issues in IBP's Domain Model was "turned off". In effect, IBP-Cases dealt with only one "issue," whether plaintiff should win the claim of trade secret misappropriation. To accommodate the single issue, modifications were made to the explaining away of counter-examples in B.2.a.ii, Fig. 2 and to the way it broadened queries in B.2.b. As shown in Fig. 7, IBP-Cases performed significantly worse than IBP (78.3 v. 91.8%); IBP-Cases abstained more frequently than IBP and made four fewer errors. However, it predicted cases correctly less often than IBP (144 vs. 169) offsetting the fewer errors. From this we concluded that the legal issues contribute to IBP's predictive accuracy. As the *National Rejectors* example in Table 1 and Fig. 5 illustrates, the legal issues focus the hypothesis- or Theory-Testing in Step B.2.a. on those Factors and conflicts among Factors that are relevant to an issue and thus help to ensure that IBP's case queries are more frequently productive than those of IBP-Cases. When IBP uses its Domain Model to construct rationales relating Factors to issues, and ultimately, to outcomes, it employs issues that judges also employ. Judges refer to the legal issues in the Restatement and Uniform Trade Secrets Act provisions, the same issues reflected in IBP's Domain Model. Thus, in formulating hypotheses about which party will win, IBP uses as its conceptual framework the "right" background knowledge about legal issues.

Even though the Domain Model and issues improve the accuracy of IBP's predictions, they are not sufficient. IBP-Model made predictions using no cases but only the issues and Factors in its Domain Model. For each issue in a new problem, IBP-Model simply tested whether all of the issue-related Factors favored the same

side. If so, IBP-Model inferred that side was favored for the issue; otherwise it abstained. As a result IBP-Model did not carry out Theory-Testing, explain away exceptions or broaden queries. As shown in Fig. 7, IBP-Model failed to perform as well as IBP, with an accuracy of only 71.7% and 38 abstentions. Although each of these abstentions involved conflicting issue-related Factors, IBP-Model could not resolve the conflicts because it could not resort to cases. For each of these abstentions, IBP's predictions were correct. Framing predictive hypotheses around issues and testing them against past cases works.

Of the three rule-learning programs, RL had the highest accuracy. It ran a close second to IBP, and the difference between them is only marginally significant. With its strategy of cover and replacement of cases, its retention of alternative rules for a training case, and its delayed, classification-phase assignment of weights to alternative rules to resolve conflicting evidence, RL seems well-suited to learning from cases like those in the CATO database that are: (1) used to resolve conflicting evidence and (2) more complex in that more than one aspect of a case may contribute to its outcome. In addition, RL's rules usually capture reasonable legal intuitions about trade secret law, and thus could serve as a basis for providing legally reasonable explanations of the predictions. By contrast, the third-ranking algorithm, Naïve Bayes, achieves quite high accuracy (although the difference is statistically significant) without using domain knowledge such as that represented in IBP's Domain Model, but cannot explain its predictions.

We would argue, however, that, by virtue of its hypothesis-testing approach, IBP explains its predictions in a natural way in terms of reasons that lawyers can assess. For each legal issue, the program formulates a hypothesis in terms of possibly conflicting Factors, finds cases that support or contradict the hypothesis, and attempts to explain away the counterexamples in terms of KO-Factors that account for their outcome and differentiate them from the positive instances. Although the concept of KO Factors would have to be explained to attorneys, all of these components of IBP's hypothesis-testing are intuitively accessible to attorneys.

We analyzed IBP's errors and found that a number of cases in the collection are anomalous in that they are very hard or even impossible to predict correctly for IBP as for most or all of the other prediction algorithms. Such anomalous cases account for more than half of IBP's errors. We found four main reasons why a case can be anomalous: (1) a case's Factor representation omitted a feature that the court deemed important, (2) although the Factor representation captured relevant facts, it failed to capture important details, (3) interpretive decisions case enterers needed to make in manually assigning Factors caused the error, or (4) the court's resolution of conflicting Factors was unique across the database of cases. For detailed examples, see (Ashley and Brüninghaus 2006). For subsequent work describing a prediction approach tolerant of noisy data, see (Wardeh et al. 2008).

It is probable that no computer model can achieve 100% accuracy for a large database of real cases realistically represented. The representations always leave something out, and even if not, real cases are sometimes remarkably balanced, the precedents conflict, and some times the precedents are wrongly decided. This also underscores the difficulty of automatically identifying a case's Factors from its text, the task of SMILE, whose evaluation is discussed next.

6 Evaluating SMILE

In a series of four sets of experiments, we evaluated the SMILE program and certain hypotheses concerning the three case text representations, BOW, RR, and ProP (Brüninghaus and Ashley 2005):

I. Which representation enabled SMILE to do the best job of assigning Factors to cases as compared to the human assignments?
II. Which representation enabled SMILE + IBP to do the best job of predicting outcomes of cases as compared to the actual outcomes?
III. How well did SMILE + IBP predict outcomes as compared to a baseline of informed guessing?
IV. How well did SMILE + IBP predict outcomes as compared to a baseline method of predicting outcomes directly from texts without using Factors?

In the first two sets of experiments, we tested each of the three text representations and measured its effect on the performance of SMILE and SMILE + IBP. Since everything else was kept constant, any observed differences in performance were attributable to the different text representations. In the last two experiments, we compared SMILE + IBP's performance against two baselines to get a better picture of how well it performed.

The intuitions underlying the three representational schemes, BOW, RR, and ProP, are straightforward. In general, our hypothesis is that representing more information about the meaning of the sentence should improve the classification process. All three representations capture only a semblance of a sentence's meaning, but each captures successively more of that meaning. As mere lists of words, BOW and RR both disregard the important contribution of word order to the sense of a sentence. RR has the apparent advantage over BOW, however, that its words are less likely to be unique to a particular case. By replacing proper names of parties and product names with terms that reflect the role in the case of the named party or the product type, we supposed, learning the classifiers should be more effective. Not only is the role information more likely to recur in other cases, it captures concepts that are important in matching. The ProP representation does not capture word order, either, but it does capture some important syntactic relationships. Since intuitively, many of the Factors focus on who did what, or what was done to what, etc., these syntactic relationships should facilitate learning the classifiers. In addition, ProPs capture some additional semantic information about the meanings of terms and the use of negation. Finally, ProPs retain the benefit of replacing names of parties and products with role information. In order to generate ProPs, we use the parser on the texts in which names have already been replaced by roles.

All three representations support mechanically comparing cases using a computationally simple way to compare sentences for purposes of determining similarity, nearest-neighbor comparison. As noted, BOW supports a computer program's comparing sentences as alpha-ordered lists of words or vectors (Jackson and Moulinier 2007, p. 30–33). In nearest-neighbor comparison, the program computes the Euclidean distance between the end points of the vectors; the smaller

the distance, the nearer (i.e., the more similar) the sentences.[26] Sentences represented in RR can be compared in the same way. Indeed, so can sentences represented as lists of ProPs, although the comparison is a bit more complex to account for the fact that there are multiple ways of expressing the same ProP given the possibility of synonyms. One simply organizes the ProPs in a kind of nested alphabetical order, treating any of the semantically equivalent versions of the ProP as a match along that dimension.

Each of the representations captures more of a sentence's meaning at the cost of increasing complexity of processing. Parsing was not an issue for either the BOW or RR representations, both of which involve mere word lists, but, the ProP representation is computationally more expensive in that it required some shallow parsing to reveal the four syntactic relationships of interest (i.e., subject—verb, verb—object, verb—prepositional phrase, and verb—adjective).

Given the above intuitions about the three representation schemes and the expected advantages and costs of representing successively more information about the meanings of the sentences, we specified two more detailed versions of hypotheses I and II:

> *Hypothesis A*: Abstracting from names and individual entities in a case text to their roles in the case allows a learning algorithm to better generalize from training examples. In other words, the prediction of this hypothesis is that the classification performance of RR will be greater than that of BOW.
> *Hypothesis B*: Using some linguistic analysis to capture (1) patterns of actions and (2) negation preserves crucial information from the text and thereby leads to better classification. In other words, the prediction is that classification with ProP will outperform that with RR.

From a scientific viewpoint, these hypotheses are important. While the limitations of a bag-of-words text representation are widely known, little research has shown the benefits of alternative approaches to represent texts in a way that makes more linguistic and legal knowledge available to guide automated decision making about the relevance and utility of the texts. From time to time syntactic phrases such as noun phrases have been added to the representation of textual documents but without demonstrable improvements in classification (Lewis 1992; Lewis and Sparck Jones 1996; Mitra et al. 1997). On the other hand, extracting more general syntactic patterns that focus on verbs had been shown to improve newswire classification precision (Riloff 1996) and has been applied to webpage categorization (Fürnkranz et al. 1998). The approach in SMILE combines the use of noun phrases and verb-based patterns.

As noted, in evaluating these hypotheses, it was pragmatically helpful to use squibs rather than the full opinions: this reduced both the length of the texts and the complexity of the sentences found in legal opinions.[27] The squib in Fig. 4 is

[26] The vectors are in a multi-dimensional space with as many dimensions as there are different words in all of the sentences (Jackson and Moulinier 2007, pp. 30–33).

[27] Jackson and Moulinier (2007, pp. 92–93) identifies a number of complexities in case opinions that bedevil information extraction: 1. The facts of the case may be intermingled with its procedural history. 2. Rulings in precedents are reported in much the same way as the ruling in the current case. 3. The opinions

representative of both the number and kinds of sentences SMILE can deal with. Our evaluation does not apply SMILE to texts as complex as full legal opinions; the squibs, however, often incorporate language taken directly from the opinions.

6.1 Experimental design

Each experiment was designed as a leave-one-out cross-validation experiment (Cohen 1995a, b). Specifically, in any given experiment, each case was removed in turn from the database and used as an input case. The SMILE program would learn each of the Factor classifiers given the database minus the input case and then would classify the text of the input case. Then, the input case was reinserted into the database, the next case became the new input case, SMILE would *relearn* all of the Factor classifiers from scratch, the new input case was classified, and so on for each of the cases in the database.

This generated a large number of experimental runs. Each of three learning algorithms (Nearest Neighbor, Naïve Bayes, and C4.5) was run with each of the three different text representations, bag-of-words (BOW), roles-replaced (RR) and propositional patterns (ProP), over all of the 26 Factors and 146 cases from the original CATO database, for a total of more than 34,000 experimental runs. Each experiment involved about 2,000 training sentences, and each input record had on average about 2,000 features. As such, the experiments took several weeks running around the clock.

6.2 Results

In Experiment I, we measured how well SMILE assigned Factors to test cases using the nearest-neighbor algorithm and each of the three text representations. The input was a case text, and the output was the set of Factors assigned to the case by SMILE. Performance was measured by comparing the set of cases to which SMILE assigned a Factor with the set of cases that had been manually marked up by human coders as having been instances of the Factor. For each Factor, we recorded the number of case assignments that were:

- correct (i.e., how many cases had the Factor and SMILE assigned the Factor).
- missed (i.e., how many cases had the Factor but SMILE did not assign the Factor).
- false (i.e., how many cases did not have the Factor but SMILE assigned the Factor).

SMILE's success in assigning a Factor, as compared to human coders, can be assessed in terms of two ratios, precision and recall, defined as follows (Jackson and Moulinier 2007, pp. 46–47):

Footnote 27 continued
contain extensive quotations from other sources. 4. The opinions may contain extensive discussions of hypothetical, counter-factual, or qualified propositions. 5. The opinions often deal with many diverse points of law.

$$\begin{aligned}\text{Precision} &\equiv \text{the proportion of relevant documents among all the documents retrieved}\\ &= (\text{correct})/(\text{correct} + \text{false}).\end{aligned}$$

$$\begin{aligned}\text{Recall} &\equiv \text{the proportion of relevant documents retrieved out of all the relevant documents}\\ &= (\text{correct})/(\text{correct} + \text{missed}).\end{aligned}$$

For convenience, precision and recall can be combined into one number, the F-Measure for a Factor.[28] The F-Measure associated with perfect precision and recall is 1.

$$\text{F} = (2 \times \text{recall} \times \text{precision})/(\text{recall} + \text{precision})$$

We then computed an average of the F-Measures for all 26 Factors.

Experiment II employed a different measure of how well SMILE assigned Factors to test cases using the nearest-neighbor algorithm and each of the three text representations. The input was a case text, and the output was IBP's predicted outcome for the case. Since the only variable is the text representation, all observed differences in prediction performance can be attributed to the representation. In this evaluation, we compared IBP's predicted outcome for a case to the real outcome of the case. We recorded the number of case predictions that were:

- correct (i.e., IBP's predicted outcome was the same as the case's actual outcome)
- abstentions or "abstains" (i.e., IBP did not make a prediction for the case)
- mistakes (i.e., IBP's predicted outcome was not the same as the case's actual outcome).

We counted the number of correct predictions, abstentions and mistakes by IBP, and then calculated its accuracy and coverage. For this purpose, we adapted the F-Measure so that we could combine these two values; we refer to it as F-Measure(Pred). Accuracy is defined as the percentage of correct predictions on those cases where IBP made a prediction and coverage as the percentage of cases where IBP made a prediction. The F-Measure(Pred) associated with perfect accuracy and complete coverage is 1.0.

$$\begin{aligned}\text{Accuracy} &\equiv \text{the proportion of predictions that were correct}\\ &= \text{correct}/(\text{correct} + \text{mistake})\end{aligned}$$

$$\begin{aligned}\text{Coverage} &\equiv \text{the proportion of instances for which a prediction was made}\\ &= (\text{correct} + \text{mistake})/(\text{correct} + \text{mistake} + \text{abstain})\end{aligned}$$

$$\text{F-Measure(Pred)} = (2 \times \text{accuracy} \times \text{coverage})/(\text{accuracy} + \text{coverage})$$

Table 4 shows the results in Experiments I and II. Table 5 shows which differences in results are statistically significant.[29]

[28] The F-Measure is the harmonic mean of the two rates, precision and recall. We used a version of the F-Measure that assigns equal weight to precision and recall. (Jackson and Moulinier 2007, p. 48).

[29] We tested the results of our experiments for statistical significance in order to show whether the observed differences were caused by true differences between the representations and algorithms, and not

Table 4 Results of experiments I and II: effect of choice of text representation on average F-Measure (Experiment I) and on F-Measure(Pred) (Experiment II)

Text representation		ProP	RR	BOW
Experiment	Measure			
I	Avg. F-Measure	0.260	0.280	0.211
II	F-Measure(Pred)	0.703	0.600	0.585

Table 5 Statistical significance of differences in experiments I and II: effect of choice of text representation on average F-measure (Experiment I) and on F-Measure(Pred) (Experiment II)

Comparison of text representations		ProP v. BOW	RR v. BOW	Prop. v. RR
Experiment	Measure			
I	Avg. F-Measure	Sig.	Sig.	NOT sig.
II	F-Measure(Pred)	Sig.	NOT sig.	Sig.

In Experiment III we compared the predictions by SMILE + IBP to a baseline of informed guessing that employed information about the predominant case outcomes in the database. Even if one had no information about which Factors apply to a test case, one could make an informed guess about the outcome based on the distribution of the outcomes in the training set. If 90% of the training cases were won by defendant, then flipping a biased coin that predicts defendant with 90% probability is the most informed prediction possible. In fact, in the CATO database 39% (146–89/146) of the cases were won by the defendant. Accordingly, the Biased-coin-flip baseline algorithm involves the following procedure: In a random experiment, predict that plaintiff wins with probability p = #-cases-won-by-plaintiff/total #-cases = 89/146 = 61%. This strategy is preferable to an alternative strategy of always predicting the majority class wins which ignores the prior probability of defendant's winning. The results were as follow:

The difference in performance was significant with $p < 0.0001$.

In Experiment IV we compared the accuracy of predictions made with SMILE + IBP and predictions made with another baseline in which the nearest neighbor algorithm was applied directly to the case texts without recourse to the Factor representation. In this leave-one-out classification experiment, SMILE + IBP used the ProP representation and the nearest neighbor algorithm, as

Footnote 29 continued
merely by chance. Because our experiments were run as cross-validations, the commonly used T-test may not lead to reliable results (Dietterich 1996; Salzberg 1997). Since the cross-validation experiments compared more than two different parameter choices (specifically, three treatments are reported here, 1 machine learning algorithm X 3 representations) we followed the procedure recommended in (Dietterich 1996). We applied a non-parametric test, Friedman's Test, to find whether there was a difference among the combinations of representations and algorithms. When this test showed significance, we used Wilcoxon's Signed-Rank test to determine whether the difference between two variants was significant (e.g., between results obtained with ProPs and Roles-Replaced using Nearest-Neighbor.) Following convention, we say that results with probability $p < 0.05$ are statistically significant, and with $p < 0.1$ marginally significant. See (Cohen 1995a, b).

described above in Sect. 3, to compare the sentences of the test case to the positive and negative sentence instances for each Factor. In the Direct-from-text prediction method, cases won by defendant were treated as positive instances of the concept "defendant-wins-trade-secret-misappropriation"; cases won by plaintiff were treaded as negative instances. Given a test case text represented as an alpha-ordered list of words, the method applied the nearest neighbor algorithm to compare it as a whole with each of the training cases so represented and then applied the classification of the nearest neighboring case. The Direct-from-text method predicted that 85% of the cases would be won by plaintiff. Its prediction that defendant wins was equally likely for cases won by plaintiff or defendant.

6.3 Discussion

In the main, the evidence from the first two experiments confirmed both hypotheses concerning the benefits of including more legal and linguistic knowledge in the text representation. Experiment I provides support for the first hypothesis (Hypothesis A); abstracting from names and individual entities in a case text to their roles in the case allows a learning algorithm to better generalize from training examples. As shown in Tables 4 and 5, Experiment I, the propositional patterns (ProP) and the roles-replaced (RR) representations both outperformed the bag-of-words (BOW) representation, each achieving a higher average F-Measure. The scores of the ProP and the RR representations were each significantly higher than that of BOW.

Focusing on the effect of text representation on predictions, Experiment II provides support for the second hypothesis (Hypothesis B); using some linguistic analysis to capture patterns of actions and negation achieves better classification. As shown in Tables 4 and 5, Experiment II, the propositional patterns (ProP) representation outperformed roles-replaced (RR) and both outperformed the bag-of-words (BOW) representation. The difference in results between ProP and RR and between ProP and BOW were both significant; the difference between RR and BOW, however, was not significant.

The results of the first two experiments were not entirely unequivocal. Experiment I did not support Hypothesis B; RR outperformed ProP, but the difference was not significant. In Experiment II, consistent with the prediction of Hypothesis A, RR outperformed BOW, but the difference was not significant.

In order to better understand the results, we investigated whether the prediction methods differed in their performance for cases won by plaintiff and those won by defendant. If a method uses legal knowledge effectively, one expects that its performance would not be affected by whether a case was won by plaintiff or by defendant. Always predicting the majority class (i.e., the side corresponding to the higher percentage of winners in the database as a whole) is an example of not using legal knowledge effectively. If plaintiffs were the majority class, such an approach always succeeds for test cases won by the plaintiff (even a stopped clock yields the correct time twice a day) but it *never* succeeds for cases won by the defendant. An attorney who behaved this way would always advise plaintiffs to sue and defendants to settle regardless of the facts of the case. By contrast, the biased coin flip baseline

has the desirable property that it performs equally well for test cases won by the plaintiff as for those won by the defendant.

In comparing the prediction performance for the three representations in Experiment II, we found that only ProP's predictive success was relatively unaffected by which side won the case. Its F-measures for the cases won by plaintiff were about the same as those won by defendant. By contrast, RR and BOW performed better for cases won by plaintiff, the majority class in our dataset, than for those won by defendant. In other words, ProP performed significantly better than RR and BOW in Experiment II, and unlike RR or BOW, its performance did not depend on which side won a case.

This more detailed investigation tends to support the conclusion that ProP is a better text representation, consistent with the implications of Hypotheses A and B. Abstracting from names and individual entities in a case text to their legal roles in the case and using some linguistic analysis to capture (1) patterns of actions and (2) negation preserves crucial information from the text and leads to better classifications.

That is not to say, however, that SMILE's classification performance, even using the ProP representation, or the prediction performance of the corresponding version of SMILE + IBP, is good. The average F-measure in Experiment I is below 0.3, which is very low.[30] It should be remembered, however, that the concepts by which SMILE + IBP categorizes the cases are specific enough to support its reasoning about the cases. At least, one can say from the results of Experiment III, Table 6 that the predictive performance of SMILE + IBP using ProPs is significantly better than an informed baseline, the biased coin flip weighted according to the probability of the majority class. The difference is actually more than it may appear. Due to the definition of the F-Measure(Pred), the apparently numerically small difference of 0.04 between F-Measure(Pred)s in Table 6 means that SMILE + IBP is 15% more accurate than the informed baseline.

Finally, SMILE + IBP was a better informed predictor than the other baseline, the direct-from-text prediction method of Experiment IV. That method, in effect, learned to predict the majority class in the database of cases. It did not use legal knowledge effectively; the prediction that defendant wins was equally likely for cases won by either side.

[30] For a variety of reasons, it is difficult to compare F-measures across the legal text categorization work described *supra* in Section 2. Not only do the categorization tasks, classification concepts, and types of legal documents differ, but so do the relative importance of recall and precision in the particular application. For a different task, automatically categorizing opinion texts by WESTLAW topic categories, (Thompson 2001) reports F-measures with $\beta = 2$, indicating that twice as much weight is given to recall as precision. (The formula is $F = ((\beta^2 + 1) \times P \times R)/(\beta^2 \times P + R)$. In evaluating SMILE, we use $\beta = 1$, treating recall and precision as having equal weight.) For different categorization methods, he reports average F-measures ($\beta = 2$) across eighteen topics ranging from .495 to .637, and for individual topics using the two best categorization methods, the measures range from .253 to .860. In other work extracting the criminal offenses (based on a standardized list) and the legal principles applied from the text of legal opinions in criminal cases, the Salomon program achieved F-measures ($\beta = 1$) of .82 and .46, respectively (Uyttendaele et al. 1998). (Daniels and Rissland 1997) do not report precision and recall or F-measures for the SPIRE program, favoring a different metric, expected search length.) It should be remembered, however, that the concepts for categorizing cases in the above work were not specific enough to support a program's reasoning about the cases as SMILE + IBP does.

Table 6 Results of experiment III: comparison of SMILE + IBP using ProP representation and biased-coin-flip baseline)

Experiment III	SMILE + IBP	Baseline (biased coin flip)
Avg. F-Measure(Pred)	0.70	0.66

There are a number of reasons why SMILE does not do better. One reason is that ProP and RR have complementary strengths, some of which are lost to SMILE when only one or the other representation is used. On average, ProP tends to work better with Factors that favor defendant; RR works better with Factors that favor the plaintiff. A number of pro-defendant Factors focus on descriptions of defendant's actions, such as F27 Disclosure-in-Public-Forum (d). This requires more information about "who did what", information captured in a propositional pattern. By contrast, several pro-plaintiff Factors focus on descriptions of situations or product features for which RR will suffice, for instance, Factor F15, Unique-Product (p). This may suggest that some combination of representations could maximize performance but raises issues of computational complexity.

Another reason that SMILE does not do better is simply that the texts are sometimes too difficult to parse, even for a then state-of-the-art shallow parser like the one we used, Sundance (Riloff and Phillips 2004). For instance, SMILE using the ProP representation missed the Factor F27 Disclosure-in-Public-Forum (d) in the *National Rejectors* case,[31] because Sundance could not accurately parse the following sentence: "The shapes and forms of the parts, as well as their positions and relationships, were all publicized in plaintiff's patents as well as in catalogs and brochures and service and repair manuals distributed to plaintiff's customers and the trade generally." Sundance mistakenly parses the verb phrase "were all publicized" as an active verb with "all" as the subject. As a result, the corresponding ProPs are of little use in assigning classifiers. When we modified the text of the sentence slightly, Sundance was able to parse it; the modified sentence was, "The shapes and forms of the parts and their positions and relationships were publicized in plaintiff's patents, catalogs and brochures and manuals, which were distributed to plaintiff's customers and the general trade." SMILE correctly identified the modified sentence as an instance of Factor F27. It matched the modified sentence to a positive instance of Factor F27 in a training case, *Dynamics*: "The first two of these features were publicized in a conference paper and an advertising brochure."[32] As parsing technology improves, one can expect improvements in SMILE's classifications.

A final reason is that some of the target concepts are simply too hard for SMILE to learn. This may be because there were too few positive instances (e.g., Factor F25 Info-Reverse-Engineered), they depend on implicit inferences from complex facts distributed across multiple sentences[33] (e.g., F8 plaintiff's information was of competitive value), they draw too subtle distinctions (e.g., F5, the nondisclosure

[31] National Rejectors, Inc. v. Trieman, 409 S.W.2d 1 (Sup. Ct. Mo., 1966).

[32] Dynamics Research Corp. v. Analytic Sciences Corp. 400 N.E.2d 1274 (Mass.App.Ct., 1980).

[33] "For current extraction technology to work, the information sought must be explicitly stated in the text. It cannot be merely implied by the text." (Jackson and Moulinier 2007, p. 106).

agreement was not specific about what it covered), they are subject to interpretive conventions that cannot be represented in SMILE (e.g., the difference between minimal security measures and none), or a combination of these causes.

7 Conclusions

The evaluation of IBP shows that a program can employ Factors not only to retrieve relevant cases and make legal arguments but to predict their outcomes with better accuracy than with most alternative prediction methods. IBP's focus on predicting outcomes issue by issue and on finding alternative hypotheses to explain away counterexamples in terms of unrelated, highly predictive Factors—not just by distinguishing them—helps provide better predictions that can be explained in terms that are intuitively meaningful to, and assessable by, attorneys.

IBP's predictive ability is limited in that it does not take into account underlying principles and policies. In addition, in Hunter's terminology all of the cases and problems "fit the model"; he distinguishes among legal domains that depend on landmark, leading, or commonplace cases; the last are more appropriate for computerized prediction because they tend to reflect the law rather than reconstruct it (Hunter 2000, pp. 54–63). The trade secret cases in CATO and IBP tend toward the commonplace. They do not address other claims, the procedural setting, or such issues as potential conflicts with federal law.[34]

The benefits of IBP's issue- and Factor-based predictions would be even more useful if Factors can be identified automatically from case texts. The work on SMILE + IBP aims at allowing programs to use information about Factors even when it is expressed in textual form. It is a small step in the direction of giving legal retrieval systems like Lexis® or Westlaw® more information they can use to help human legal researchers solve problems. Ideally, a SMILE-type program might process the case texts a legal IR system retrieves in response to user queries, highlighting Factors and issues relevant to the user's problem.[35] Given a Factor representation of the user's problem, or generating it from the user's textual description, SMILE could enable IBP to help the user pose and assess hypotheses about how the problem should be decided using the cases retrieved from the legal retrieval system that SMILE has also translated into Factors.[36]

Unfortunately, the accuracy of SMILE's classifications is much too poor to realize this goal. Perhaps the most that can be said is, like Samuel Johnson's

[34] These issues may interact with issues IBP's model does address. For instance, if a defendant copied information fixed in a tangible medium of expression and covered by the subject matter of copyright, a trade secret claim may be preempted under s. 301 of the Copyright Act. A trade secret claim that involved an extra element of breach of confidence would not be preempted, but it could be if it involved only improper means. Although IBP's model does not address preemption, some of the same Factors (e.g., regarding confidential relationship) would be relevant to the preemption analysis. One would need to modify the model to add preemption.

[35] Currently, a legal IR system ranks the cases it retrieves according to statistical criteria that involve the frequencies of the query's terms' appearances in the retrieved cases and in the corpus.

[36] Alternatively, a SMILE-type system might automatically classify cases drawn from legal IR systems and index them in a specialized database.

description of a "dog's walking on his hind legs. It is not done well; but you are surprised to find it done at all." (Boswell et al. 1988). Nevertheless, the fact that the combined SMILE + IBP system can predict and explain the outcomes of case facts *input as texts* is a milestone in the field of AI & Law; it marks the first time to our knowledge that a program can reason automatically about legal case texts. SMILE has been shown to perform better than some plausible baseline approaches, and its techniques for improving upon the commonly used bag-of-words text representation by including some legal and linguistic knowledge in its role replacements and propositional patterns have been shown to improve learning and performance. In addition, the SMILE program works well enough that its companion IBP can reason with the cases so represented, predict their outcomes and explain the predictions. The effect of SMILE's classifications on these predictions has been employed as a yardstick for measuring its performance.

As such, SMILE + IBP should be seen as a kind of existence proof; it demonstrates the feasibility of a program's reasoning about legal cases input as texts, while at the same time identifying various limitations that would need to be addressed before the approach could actually be coupled with a more traditional legal information retrieval system. First, SMILE deals only with one kind of legal claim, trade secret misappropriation, but its methodology should apply in any legal domain where courts note stereotypical patterns of fact that tend to favor one outcome over another in a legal claim or theory.[37] Second, it assumes that a technique is available automatically to substitute role information for the parties' and product names in the texts, so that the classifier can "learn" from more general examples. We have developed automated techniques for extracting role information, and shown that they work reasonably well, but it is likely that other better proprietary techniques are available. It also assumes that common phrases, important in particular legal domains, can be identified automatically. Third, the textual descriptions SMILE deals with have been manually abstracted from legal case opinions and are on the order of a few paragraphs long, much shorter than full-text legal opinions. The texts all contain mostly fact descriptions, unlike full opinions where factual descriptions may not be clearly delineated from discussions of law, application of law to facts, or facts of other cases. As natural language parsing techniques improve, and as programs improve in differentiating parts of an opinion, the accuracy of automatically classifying full legal opinions will improve, as well.

We expect that future researchers will address the problem of automatically classifying case texts by Factors differently, perhaps seeking to exploit techniques for unsupervised machine learning to leverage the small quantities of manually annotated examples that were available to us. One might begin with a very large set of case opinions dealing with trade secret law. Argument mining techniques might be employed to focus on those portions of the opinions where the court is stating the

[37] This AI & Law research on automatically processing case texts assumes that the opinion texts are reasonably complete and candid descriptions of courts' decisions. SMILE + IBP's focus on Factors for and against a court's decision, for example, assumes that courts adequately disclose the facts in a case that favor a contradictory decision, a point about which legal scholars differ. See (Delgado and Stefancic 2007, fn. 100).

facts of the case, that is, the locations where the court is most likely to make statements from which the applicability of Factors can be inferred (Moens et al. 2007). These would constitute the unlabeled examples. In addition, a small set of manually classified instances like the ones we used would be introduced as labeled seeds. Then some technique for bootstrapping, weakly supervised learning, or active learning with clustering, would be applied to learn the varying classification patterns and maximize predictive accuracy (Moens 2006, ch. 6).

The ultimate goal, however, would be the same as in SMILE + IBP, to bridge extracting information from case texts and case-based reasoning, to extract information from textual descriptions of the facts of decided cases and apply that information to predict and explain the outcomes of new cases.

Acknowledgment The research described here has been supported by Grant No. IDM-9987869 from the National Science Foundation.

References

Aleven V (1997) Teaching case-based argumentation through a model and examples. Ph.D. dissertation, University of Pittsburgh

Aleven V (2003) Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment. Artif Intell 150:183–237

Ashley K (1988) Modeling legal argument: reasoning with cases and hypotheticals. Ph.D. dissertation. COINS technical report No. 88–01, University of Massachusetts, Amherst

Ashley K (1990) Modeling legal argument: reasoning with cases and hypotheticals. The MIT Press, Cambridge

Ashley K (2000) Designing electronic casebooks that talk back: the CATO program. Jurimetrics J 40:275–319

Ashley K (2002) An AI model of case-based argument from a jurisprudential viewpoint. Artifi Intell Law 10:163–218

Ashley K, Brüninghaus S (2006) Computer models for legal prediction. Jurimetrics J 46:309–352

Bench-Capon T, Sartor G (2001) Theory based explanation of case law domains. In: Proceedings of the 8th international conference on artificial intelligence and law. ACM Press, pp 12–21

Bench-Capon T, Sartor G (2003) A model of legal reasoning with cases incorporating theories and values. Artif Intell 150:97–143

Boswell J, Chapman R, Fleeman J, Rogers P (1988) Life of Johnson. Oxford University Press, Oxford

Branting L (1999) Reasoning with rules and precedents—a computational model of legal analysis. Kluwer, Dordrecht

Brüninghaus S, Ashley K (1999) Bootstrapping case base development with annotated case summmaries. In: Proceedings of the third international conference on case-based reasoning. LNAI 1650. pp 59–73

Brüninghaus S, Ashley K (2001) Improving the representation of legal case texts with information extraction methods. In: Proceedings of the eighth international conference on artificial intelligence and law. pp 42–51

Brüninghaus S, Ashley K (2003) Predicting the outcome of case-based legal arguments. In: Sartor G (ed) Proceedings of the 9th international conference on artificial intelligence and law (ICAIL-03). ACM Press, pp 234–242

Brüninghaus S, Ashley K (2005) Reasoning with textual cases. In: Proceedings of the sixth international conference on case-based reasoning. Springer, pp 137–151

Burke R, Hammond K, Kulyukin V, Lytinen S, Tomuro N, Schonberg S (1997) Question answering from frequently-asked question files: experiences with the FAQ finder system. 18 Ai Magazine 18:57–66

Cardie C, Howe N (1997) Improving minority class prediction using case-specific feature weights. In: Proceedings of the fourteenth international conference on machine learning. Morgan Kaufmann, pp 57–65

Chapman W, Bridewell W, Hanbury P, Cooper G, Buchanan B (2001) A simple algorithm for identifying negated findings and diseases in discharge Summaries. J Biom Inform 34:301–310, 302
Chorley A, Bench-Capon T (2005) An empirical investigation of reasoning with legal cases through theory construction and application. Artif Intell Law 13:323–371
Cohen P (1995a) Empirical methods for artificial intelligence. MIT-Press, Cambridge, MA
Cohen W (1995b) Text categorization and relational learning. In: Proceedings of the twelfth international conference on machine learning, pp 124–132
Cunningham C, Weber R, Proctor J, Fowler C, Murphy M (2004) Investigating graphs in textual case-based reasoning. In: Proceedings of the seventh european conference on case-based reasoning, pp 573–586
Daelemans W, Zavrel J, van der Sloot K, van den Bosch A (2004, 2007) TiMBL: Tilburg Memory Based Learner, version 5.02 (now 6.0) http://ilk.uvt.nl/timbl/
Dale R (2000) Handbook of natural language processing. Marcel Dekker, Inc., New York
Daniels J, Rissland E (1997) Finding legally relevant passages in case opinions. In: Proceedings of the sixth international conference on artificial intelligence and law. ACM Press, pp 39–46
Delgado R, Stefancic J (2007) Why do we ask the same questions? The triple helix dilemma revisited. Law Libr J 99:307–328
Dietterich T (1996) Statistical tests for comparing supervised classification learning algorithms. Oregon State University Technical Report
Ejan Mackaay E, Robillard P (1974) Predicting judicial decisions: the nearest neighbour rule and visual representation of case patterns. Datenverarbeitung im Recht 3:302
Fürnkranz J, Mitchell T, Riloff E (1998) A case study in using linguistic phrases for text categorization on the WWW. In: Proceedings of the ICML/AAAI-98 workshop on learning for text classification. Technical Report WS-98-05, pp 5–120
Gonçalves T, Quaresma P (2005) Is linguistic information relevant for the text legal classification problem? In: Proceedings of the tenth international conference on artificial intelligence and law. ACM Press, pp 168–176
Gordon T, Prakken H, Walton D (2007) The carneades model of argument and burden of proof. Artif Intell 171:10–11
Grover C, Hachey B, Hughson I, Korycinski C (2003) Automatic summarisation of legal documents. In: Proceedings of ninth international conference on artificial intelligence and law. ACM Press, pp 243–251
Hachey B, Grover C (2006) Extractive summarization of legal texts. Artif Intell Law 14:305–345
Hanson A (2002) From key numbers to keywords: how automation has transformed the law. Law Libr J 94:563
Hunter D (2000) Near knowledge: inductive learning systems in law, Virginia. J.L. & Tech. 5:9
Jackson P, Moulinier I (2007) Natural language processing for online applications: text retrieval extraction and categorization, 2nd edn. John Benjamins Publishing Co, Amsterdam
Jackson P, Al-Kofahi K, Tyrrell A, Vacher A (2003) Information extraction from case law and retrieval of prior cases. Artif Intell 150:239–290
Kim Won, Wilbur W (2000) Corpus-based statistical screening for phrase identification. J Am Med Inform Assoc 7:499–511
Lenz M (1999) Case retreival nets as a model for building flexible information systems Ph.D. dissertation, Humboldt University, Berlin
Lewis D (1992) Representation and learning in information retrieval. Ph.D. dissertation, University of Massachusetts, Amherst
Lewis D, Sparck Jones K (1996) Natural language processing for information retrieval. Commun ACM 39:92–101
McCallum A (2004) Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/_mccallum/bow
McCarty LT (2007) Deep semantic interpretations of legal texts. In: Proceedings of the eleventh international conference on artificial intelligence and law, pp 217–224
Mitchell T (1997) Machine learning. McGraw-Hill, New York
Mitra M, Buckley C, Singhal A, Cardie C (1997) An analysis of statistical and syntactic phrases. In: Proceedings of the fifth international conference “recherche d’Information assistee par ordinateur”, pp 200–214
Moens M-F (2006) Information extraction: algorithms and prospects in a retrieval context. Springer, Dordrecht

Moens M-F, Boiy E, Palau R, Reed C (2007) Automatic detection of arguments in legal texts. In: Proceedings of eleventh international conference on artificial intelligence and law (ICAIL-07), pp 225–236

Popple J (1996) A pragmatic legal expert system. Dartmouth. Ashgate. Farnham, UK

Provost F, Aronis J, Buchanan B (1999) Rule-space search for knowledge-based discovery. CIIO Working Paper #IS 99-012, Stern School of Business, New York University (visited March 23, 2009) <http://pages.stern.nyu.edu/~fprovost/Papers/rule-search.pdf>

Quinlan R (1993) C4.5: programs for machine learning. Morgan Kaufmann, San Francisco

Quinlan R (2004) C4.5 Release 8. http://www.rulequest.com/Personal/

Riloff E (1996) Automatically generating extraction patterns from untagged text. In: Proceedings of the thirteenth national conference on artificial intelligence, pp 1044–1049

Riloff E, Phillips W (2004) An introduction to the sundance and autoslog systems, University of Utah School of Computing Technical Report #UUCS-04-015. http://www.cs.utah.edu/~riloff/pdfs/official-sundance-tr.pdf (visited March 23, 2009)

Rose D (1994) A symbolic and connectionist approach to legal information retrieval. Lawrence Erlbaum Publishers, Taylor & Francis Group, Philadelphia

Salzberg S (1997) On comparing classifiers: pitfalls to avoid and a recommended approach. Data Min Knowl Disc 1(3):317–328

Thompson P (2001) Automatic categorization of case law. In: Proceedings of the eighth international conference on artificial intelligence and law. ACM Press, pp 70–77

Turtle H (1995) Text retrieval in the legal world. Artif Intell Law 3:5–54

Uyttendaele C, Moens M-F, Dumortier J (1998) SALOMON: automatic abstracting of legal cases for effective access to court decisions. Artif Intell Law 6:59–79

Vossos G (1995) Incorporating inductive case-based reasoning into an object-oriented deductive legal knowledge based system. Ph.D. dissertation, Latrobe University, pp 146, 157

Wardeh M, Bench-Capon T, Coenen F (2008) Argument based moderation of benefit assessment. In: Legal knowledge and information systems, Proceedings, Jurix 2008: The twenty-first annual conference, pp 128–137

Weber R (1998) Intelligent jurisprudence research. Doctoral dissertation. Federal University of Santa Catarina, Florianópolis, Brazil

Witten I, Eibe F (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco

Zeleznikow J, Hunter D (1994) Building intelligent legal information systems–representations and reasoning in law. Kluwer, Amsterdam