Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

Automatic Categorization of Statute Documents

Tom Curran
Paul Thompson
West Group
Eagan, Minnesota 55123
tpcurran@westpub.com
thompson@research.com

1.0 Introduction

Automatic classification offers publishers of large document collections the possibility of improved production efficiencies in print and online environments. In this paper we explore the possibility of automating the classification of statutory legal materials through the application of machine learning software designed to generate automatic text categorization. Our investigations focus on a specific methodology. Our plan aimed to train classifications from a pre-classified dataset of statute documents and associated index references. Accordingly, we observed that each index feature[1] like 'insurance', or 'corporations' appended a set of document locators[2]. These locators make up the *local collection* for that index feature. The total of all documents in the dataset, whether assigned an index feature or not, makes up the *global collection.* The fundamental idea was to develop an algorithm based on text features whose frequency in the local collection was high but whose frequency in the global collection was moderate to low. The system would be provided with a set of descriptors taken from the text of statute documents from which it generates, by algorithm, a lexicon.[3] The lexicon is evaluated by domain experts who assess its relationship to the semantic content of the index feature sought to be modeled. Once a satisfying lexicon has been created, machine learning software is used to generate classification rules from the lexicon. The rules in turn generate classifications for documents in a test collection.

Our experiment consisted of two basic steps: (i) creating a semantically satisfying model from text features occuring in each distinct local collection which would adequately represent the index feature associated with that collection, and (ii) using the model to automatically classify statute documents. Specifically, once models were created, they would be used as an input list of attributes for automatic text categorization software. The software would make rules, or in some cases decision trees based on analysis of a training corpus of statute documents. These rules could then be used to classify statutes from a test dataset. The results would be output in a "confusion matrix" based on agreements and disagreements of human classifiers and the machine-learning software.

In the discussion that follows, we clarify the nature of statute documents and associated organizational schema; then we describe the statute document collection and related index corpus in terms of our efforts to replicate parts of the human indexing process by breaking it down into rule-governed phases; we articulate the differences between index types as they apply to statute collections and discuss the relationship of automatic text categorization to our data; and finally we discuss results and conclusions to be drawn from our work.

2.0 Statutory Collections

[1] The topics we address are cross-disciplinary, involving indexing, classification science, information retrieval, text-categorization and linguistics. To be as certain as we can be that ambiguity has been avoided we use 'index feature' as a synonym for 'index classification', 'index term', and 'index descriptor'. Similarly, strings occurring in text documents shall be referred to as 'text features'.

[2] In ordinary free text, locators are customarily page numbers. In the instant case, locators are legislatively supplied unique document identifiers attached to sections of codified legislation. Index locators are citations to statute locators following the text portion of a reference.

[3] Again, in an interdisciplinary environment, the set of text features chosen to represent a given index feature might variously be called 'lexicons', 'models', 'profiles', or 'attributes'. We use 'model'.

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

Statutes differ significantly from other legal documents in that they are hierarchically organized, as shown in figure 1.

Level 1 (Collection name)	**MASSACHUSETTS GENERAL LAWS ANNOTATED**
Level 2 (Sub-collection)	**PART I. ADMINISTRATION OF THE GOVERNMENT**
Level 3 "	**TITLE XXI. LABOR AND INDUSTRIES**
Level 4 "	**CHAPTER 149. LABOR AND INDUSTRIES**
Level 5 "	**GENERAL PROVISIONS AS TO EMPLOYMENT**
Level 6 "	**s 19A. Copy of medical report for employee**
Level 7 (text level)	**Any employer requiring a physical examination of an employee shall, upon request, cause said person to be furnished with a copy of the medical report following the said examination.**

Figure 1. Statute Document with Legislative Hierarchy

Notice that the text of the statute shown carries a set of repeated hierarchical lines associated with each unit of located text. These lines, which occur throughout a given statute collection are known to legal practitioners as a "legislative hierarchy" or "statutory scheme". These examples, from West Group's online service, show hierarchies repeated for each bottom-level hierarchical text unit (called a "statute section").

3.0 Preparation of Document Collections; Statute and Index Corpora

3.1 *The document collection.*

For the initial stage of our investigations, we chose a sub-collection of previously classified statutory documents. The sub-collection contained 149,655 documents. An average document was classified 8.4 times. Documents were normalized by programmatic removal of irrelevant chunks of text wherever possible. This meant, in most cases, the removal of special editorial enhancements and composition markup. We tried to remove all strings not specifically related to content. Normalized documents consisted of a "begin-doc" marker, hierarchy markers with associated strings, caption markers with associated strings, text markers with associated statute text paragraphs and document locators consisting of unique, specially formatted strings.

3.2 *Index features; the index master corpus.*

Associated with most documents in the document collection was a set of index references containing one or more citations to associated documents in the local collection. Each reference itself contained at least two and often several distinct classifications. As we mention elsewhere, hierarchical organization creates topics and sub-topics that are bound to each other through each hierarchy. This means that multi-term references, as opposed to uniterm references, must be used to express full statute content. For example, statutory provisions on state government include provisions relating to state employees which, in turn may

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

include provisions relating to health insurance. If the text at the bottom of this hierarchy pertained to premiums, the referential model for an index entry to these nested topics might appear:

STATE
Employees, health insurance, premiums, **Ch. 10, s. 125**

References were disassembled into separate index features and compiled into a lookup file containing all such features and associated document locators. In the case of the foregoing example, the final table of index features may be represented thusly:

Index Feature	**Locator**
employees	ch. 10, s. 125
...	
health insurance	ch. 10, s. 125
...	
premiums	ch. 10, s. 125
...	
state	ch. 10, s. 125

Figure 2. Index Master Corpus

This is the index master corpus. Index locators are identical in form and content to locators used to identify associated statutes in the text document collection. Routing connections between index features and associated statute text documents are illustrated in Figure 3.

Index Feature => Index Locator => Statute Document Locator => Statute document

Figure 3. Routing connectivity between Index Features and Statutes Text

Relations between index features and text documents may be one-one, one-many, many-one, or many-many

3.3 *Indexing strategies; creation of the master corpus.*

Domain experts indexed the subject collection over a period of more than a decade. Each expert was a lawyer and received specific prior training in the assignment of index features to candidate documents using an established control vocabulary of approximately 1000 index features. The manual indexing process can be broken down into three phases.

In the first phase, humans list all index features from the control vocabulary which topically describe the content of the subject statute document. This phase, it should be noted, occurs only in the abstract. The indexer notes text features upon which classification is to be predicated and makes a mental list of candidate classifications from the index feature control vocabulary[4].

In the second phase, indexers eliminate from the first-phase list index features not well suited to a print environment. In general, there are two reasons for eliminating first-phase assignments: (1) to limit

[4] It is the absence of a physical list that necessitates review of machine-classified documents for "true noise" and "false noise". See the discussion of false noise in section 8.2.

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

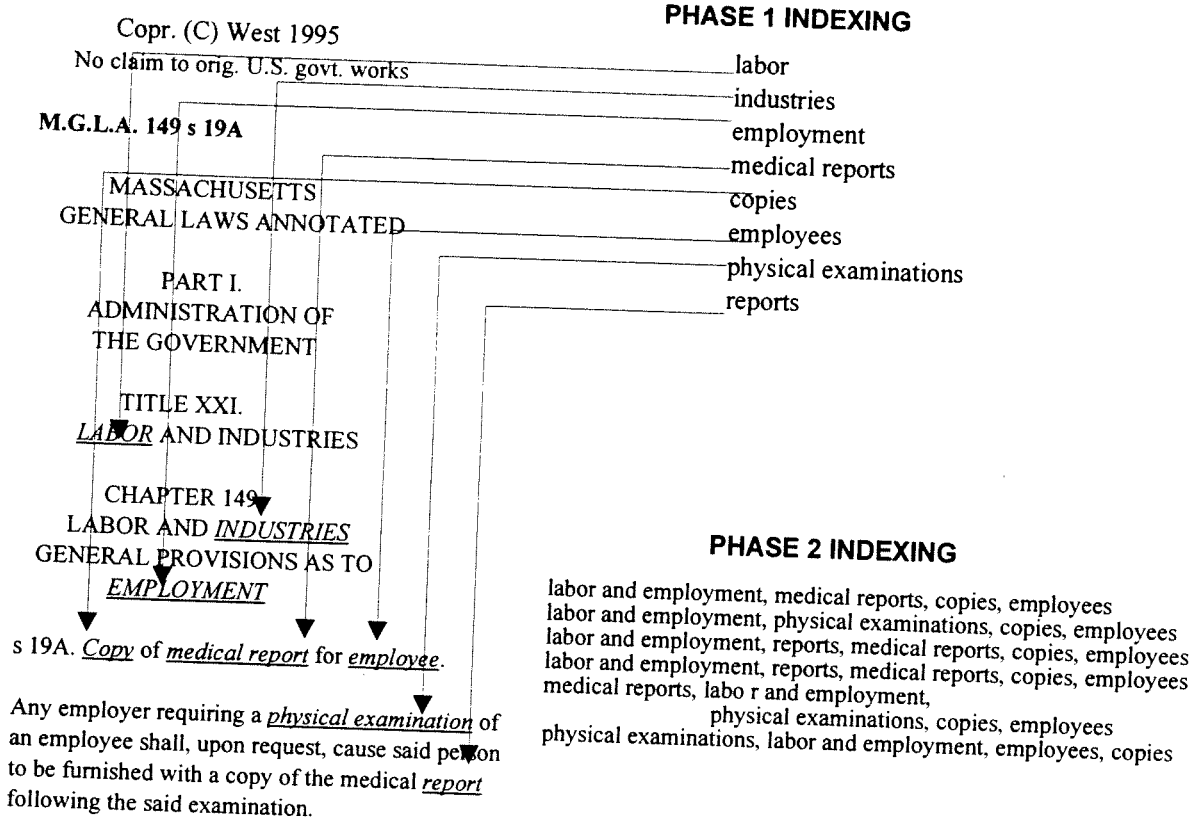

Figure 4. The Three Phases of Statute Indexing

exhaustivity[5] and (2) to avoid print redundancy by eliminating topics that are too broad to index, like "Administration of Government" and synonyms like 'Industries', which gets represented in the print index by a cross-reference pointer to 'LABOR AND EMPLOYMENT'.

[5] The number of classifications made to a particular document. Fully exhaustive classification tends to destroy or minimize features of an index designed to emphasize the relative importance of certain topics in relation to others of lesser importance. For example, if a document contains a single reference to felony-murder along with 20 references to copies of trivial reports, the felony-murder topic is made more difficult to find in a print product because more entries must be perused to locate it.

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

In the final phase, index features remaining after phase-2 elimination are concatenated to create multi-term references, which are in turn collated and compiled to create a finished index. Statute print indexing is the application of the three-stage indexing process to statute documents: the selection and assignment of index features to represent selected topics from full statute sections and associated hierarchies, assembling the terms into sequences called "references" attaching locators (which, in statutes are section-numbers rather than page numbers) and normalizing the result. Figure 4 above shows phase-1, phase-2 and phase-3 indexing.

Investigators were concerned primarily with recreating the first, classificatory stage of the indexing process through machine learning though they acknowledge the effect the second stage has on consistency of classification. The reason for limiting machine-classification to phase-1 indexing may be expressed as the difference in rules governing phase-1 and phase-2 classifications, where {X }is a set of text features which promote the assignment of classification C to a document, *n*:

Phase-1 rule: Assign C to *n* if and only if {X} occurs in *n*;

Phase-2 rule: Assign C to *n* only if {X} occurs in *n*.

The phase-1 rule specifies that a yes-no classification decision must be made accordingly as certain words or phrases, {X} do or do not occur in a target document. The phase-2 rule provides only that for a yes-classification to be made, {X} must be present, leaving the discretion to apply a no-classification to the indexer even if {X} occurs in the document.

4.0 Experiments in Automatic Statutes Classification

4.1 *Derived and Assignment Indexes.*

Automating strategies divide accordingly as index classifications are derivative or by assignment. Derivative indexing selects index features from the text features occurring in candidate documents. Assignment indexing selects and assigns index features to statute documents which, frequently, do not appear in the text of the classified documents. A preference for assigned or derived indexing in large document collections appears to hinge on whether or not the target collection is hierarchically organized and whether or not it is representable by a consistent stock of control-vocabulary terms. Where schematic hierarchies range over and relate many full-text documents, schematically expressed concepts may be assumed to be of great importance. When these same concepts appear in hierarchically remote portions of the collection, the most straightforward way to relate them back to the original schematic representation is by tight vocabulary control.

4.2 *Modeling index features from training data.*

Training from data amounts to using legacy documents to which an index feature has been previously assigned to create a textual model for that feature. The textual model consists of a ranked text feature list which is then machine-compared to documents in an unclassified or test collection. Documents resembling the model are classified by assigning the modeled index feature to them. Processing and evaluating textual models was done with machine-learning software.

4.3 *Recent History of Automatic Text Categorization.*

Automatic categorization of text based on previous manual categorization has been the topic of a variety of recent research projects. As discussed by Lewis and Ringuette (1994) this research has tended to follow two main approaches. The first uses human knowledge engineering to build a rule-based system, much like an expert system, for categorization. Among the most accurate results are those obtained by CONSTRUE, a rule-based system, using manually encoded rules to categorize 723 incoming newswire text into 674 economic and financial news categories (Hayes 1992). While highly effective, the development of this system reportedly took approximately four person-years. The second approach, requiring much less development time, but with less accurate results, uses machine learning. One machine learning approach based on a nearest neighbor algorithm is memory-based reasoning. Creecy et al. (1992) describe a system that automatically categorizes U. S. Census Bureau text into 232 industry codes (71% accuracy) and 504 occupation codes (62% accuracy). Masand et al. (1992) discuss another memory-

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

based reasoning application where incoming newswire text is categorized into 361 categories. Other recent machine learning approaches to text categorization include Lewis and Gale (1994), Apte et al. (1994), Lewis et al. (1996), Cohen and Singer (1996), Hodges et al. (1996), Moulinier et al. (1996), and (Leung & Kan 1997). The experiments described in this paper also follow a machine learning approach.

For purposes of the instant experiment the document collection and master index corpus were segregated into a training corpus of 125,180 documents and a test corpus of 24,475 . Both collections had previously been classified by human experts using index features from the standardized control-vocabulary. The index corpus contained approximately 1,250,000 classifications. Machine classifications to test documents were validated by comparing output from the machine-learning software to classifications of the same documents made previously by domain experts. This methodology was chosen after preliminary investigations using other methods, described more fully in paragraphs 4.4 and 4.5 either failed or appeared to show little promise.

4.4 *Direct similarity comparisons of text documents.*

A preliminary study was conducted which simply compared statute documents from training and test collections, in hope that enough similarity could be found between training and test documents to "poach" classifications of test documents from classifications already made to training documents. This procedure failed for all but a few test documents.

4.5 *Standard inverse document frequency measures.*

The investigators also concluded that standard inverse document frequency (IDF) measures for evaluating candidate index features did not work well with statute text. Statute collections, hierarchically organized, appeared to require multi-term indexing, since the hierarchies more often than not reflected topical subdivisions among text documents[6]. Since the hierarchies often contained text terminology highly useful in modeling candidate index features, the collection frequencies of these terms tended to be high because each document at the lowest structural levels in the hierarchies inherited all hierarchical occurrences of the candidate terms. The natural, statistical distribution of text features that might support the porting of index features from training to test collection appeared to require an algorithm other than simple IDF.

5.0 Experimental hypotheses

5.1 *Partition of the training corpus into local and global collections.*

Investigators observed that each text feature in the document collection to be modeled could be viewed in two ways: globally, in terms of its frequency in the collection as a whole, and locally, in terms of its frequency inside a given topical sub-collection. They hypothesized that text features in training data, which appeared frequently inside a local collection but less frequently outside that collection, would support machine assignment of an appropriate index feature from the training data to the document in the test collection. Figure 5 on the following page shows a partial set of text features for the index feature "insurance".

[6] See, e.g., Figure 4.

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

Text Features (stemmed)	Global Document Frequency	Local Document Frequency
binder	81	48
intestin	14	8
underwrit	568	33
reversionar	53	31
tornado	19	11

Figure 5. Selected Text Features Associated with the Index Feature *Insurance*

5.2 *Semantically nuanced language; triage.*

Investigators further hypothesized that statutory language was in many cases, heavily nuanced, so that preliminary extraction of text features according to an algorithm devised to represent sub-collections semantically, as far as possible, would yield the best set of terms to model a given index feature. Domain experts were the ultimate judges of whether a particular algorithm yielded, or failed to yield a semantically satisfying set of text features. Where models were not altogether satisfying, reviewing experts modified feature sets to by removing obvious noise and adding terms they felt would improve semantic content.

Investigators expected that results would triage to form: (1) a small but significant set of index features classifying target statute documents with high precision and recall "out of the box" with no need of further improvement; 2) another, larger set of index features that could be modeled and subsequently brought to an acceptable level of performance only by refinements added to the model by a combination of specialized parameter settings and intelligent intervention by domain experts; and (3) another, smaller set that would perform poorly and resist improvement altogether. This hypothesis projected that modeling would succeed accordingly as models employed statute text features which appeared in most local statute documents assigned to a particular index feature in the training data and appeared in few of the documents outside the local collection. As language distribution in training documents assigned to a particular index feature tended to be indistinguishable from language in documents *not* assigned that feature, the index feature in question could not be successfully modeled.

6.0 Machine Learning Text Categorization Software

Quinlan's C4.5 algorithm (1993) is a machine learning decision tree approach to automatic categorization. Through training data the system learns rules to categorize new input data. To use the C4.5 algorithm it is necessary to select features, or, in the terminology of this paper, text features, with which to represent each document. In the automatic document categorization literature feature selection algorithms are described which: a) select a single set of features for all documents, e.g., Lewis and Ringuette (1994); b) select a separate set for each category, or index feature, e.g., Apte et al. (1994) or Moulinier (1997). For statistical algorithms, including, at least to some degree, machine learning algorithms such as C4.5 which are statistically based, the number of training documents available determines how many features, i.e., text features, can be used. For these experiments we have 125,180 training documents, which, at 75 documents per feature would allow 1,669 features (Lewis 1992). So far, typically anywhere from 60 to 200 features have been used. The largest number of features was 783. For machine learning one of the main feature selection considerations is computational complexity reduction, so that programs do not run out of memory. C4.5 can be used with multiple categories directly, but more typically in text categorization research if there are n possible categories, the problem is broken up into n separate binary categorization problems. For example, is this document an insurance document, or a non-insurance document?

The C4.5 algorithm produces a decision tree with which to categorize documents. Optionally the decision tree can be converted to a set of rules, which also categorizes the documents. The categorization provided by rules is usually more accurate than the tree result and also easier for a person to understand. A decision tree is created by first identifying the single text feature, and a test associated with the feature, which is most informative in terms of categorizing all training documents into the target category, or the non-target category, based on the information gain ratio. The test is whether or not the value of the feature in the training document is greater or less than or equal to a certain value. The information gain ratio is a normalized version of the mutual information between the category and the feature.

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

The first feature selected becomes the root of the decision tree. A branch is taken for each document being categorized according to whether or not the test is passed. In a similar way additional features and tests are selected. If at any point insufficient information is gained by performing a further test, no further branching takes place and that node becomes a leaf. All documents that reach the leaf are categorized accordingly. The tree that has been created may not categorize all training documents correctly, but it is closely fitted to the training examples. The danger is that it may be overfitted so that the tree will not perform well on test data. This tree, called the unpruned tree, is then pruned to produce a tree that is less accurate on training data, but is expected to be more accurate on test data. Every decision tree can be converted to an equivalent set of rules. If a rule set is generated, the C4.5 algorithm first makes the set of rules equivalent to the unpruned decision tree. The rule set is then pruned in a way similar to the way the decision tree was pruned in an attempt to produce better categorization.

The C4.5 algorithm is designed to be used with a training set and a test set of documents. The C4.5 algorithm optimizes error rate, which is not necessarily a good criterion for text categorization, since error rate is based on the number of miscategorizations, which gives equal weight to recall and precision. If one or the other of recall or precision is considered more important, then optimizing error rate is not a good criterion. This weakness of C4.5 has been corrected in C5.0 the successor to C4.5, which has recently become available (Quinlan 1997). C4.5 can be run with a variety of tree and rule parameter settings. In an earlier categorization project two tree parameters were adjusted. The "*m*" parameter controls the shape of the decision tree generated by C4.5. The intent is to prevent the proliferation of "near-trivial" tests where almost all of the training cases have the same outcome. Using "*m*" requires that each test lead to at least two branches with at least "*m*" outcomes on each branch. The default setting is *m* = 2. The "*c*" parameter is a confidence level controlling decision tree pruning. Its default setting is 25%. Various combinations of these two parameter settings were used in an earlier project. Since the default setting, where *m* =2 and c = 25, performed, on average, about as well as any of the other settings, all runs on the statutes indexing project used the default setting for trees. Some runs were done using the "*c*" parameter with a setting of 50 for rules. This setting caused less pruning of the rules and was used to try to overcome the tendency on some categories to simply classify all documents to the default category, i. e., the non-target category.

7.0 Results

The investigators are continuing to experiment with new feature selection and weighting algorithms, but have established baseline results for 36 of the original 37 index features. One index feature, *Personal Property Taxation*, was removed because it is archaic. Baseline runs used a mixture of single-term and phrase features with no stemming. Table 1 shows results for 5 of the 36 index features interpreted in accordance with preferences expressed by domain experts to weight precision more heavily than recall.

For proposed standards of success, classification experts suggested that a combined precision-recall measure in the 80-90 % range would be considered workable. Here the combined measure being used is the F-measure with β = 0.5. The F-measure, due to van Rijsbergen (1979), combines precision and recall in a single measure with more or less weight being given to recall or precision, depending on the value of β. With β = 0.5 twice as much weight is given to precision as compared to recall. A combined measure of over 50% is considered improvable, and a combined measure of 50% or less would be considered recalcitrant. Results under these standards tend to support the triage hypothesis. The three strongest categorizations were 'Warehouse Receipts', 'Insurance', and 'Workers' Compensation', which tended to have fully developed trade languages, replete with identifiable nuance. Among the weaker performers were 'Counties', 'Sales', and, suprisingly, 'Probate Proceedings' which tended to be couched in vanilla language statistically indistinct from the vocabulary of everyday natural language. Table 2 shows averages and standard deviations for recall, precision, and F-measure scores with β = 0.5 for the 36 index features.

Precision results are based on 35 index features. The results for the index feature, *Easements*, are not included because for the category, all documents were categorized as being non-*Easements* documents. Thus precision was undefined and could not be included in the average. This result for *Easements* illustrates the C4.5 error rate optimization problem described in section 6. Because such a small proportion of the collection is *Easements* documents, C4.5 achieves a low error rate simply by categorizing all documents as non-*Easements* documents.

			F-measure

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

Index Feature	Precision	Recall	$\beta = 0.5$
Counties	48.09	12.29	30.39
Sales	73.68	13.00	38.11
Worker's Compensation	78.71	91.73	81.01
Insurance	83.19	84.56	83.46
Warehouse Receipts	92.59	86.21	91.24

Table 1. Results for Selected Index Features

Precision $n = 35$	σ	Recall $n=36$	σ	F-measure $\beta = 0.5$	σ
67.62	17.43	41.61	23.50	56.35	18.92

Table 2. Averages for all Index Features

8.0 Discussion

8.1 *The categorization algorithm.*

A machine learning algorithm can perform categorization only based on the features and feature values with which it is presented. Some argue that for text categorization feature selection is relatively unimportant, because if, say all non-stop words in the document collection are used as features, the algorithm itself will select the best features when it creates trees or rules. This may be true if available CPU performance and RAM capacity can handle such large feature sets, but if not, feature selection is critical. Furthermore, features other than single words, or word stems, such as phrases, or collocations, may give better performance. Collocations are words appearing within a certain distance of each other, but not necessarily as a phrase. Feature sets based on phrases or collocations may not perform as well as expected, however, because of their relative infrequency within training and test documents.

Categorization performance varied widely from category to category in this study. To some extent this was a function of the number of positive examples in the training data available for a given category, but some categories seemed to be more difficult to predict even when the number of positive training examples was taken into account.

8.2 *False noise.*

Results were further affected by the discovery of "false noise" generated by legacy data. In evaluating results, investigators looked at sample documents from each cell of a confusion matrix. With respect to a given index feature, C, and document, *n*, the matrix contains four sets of outcomes accordingly as domain expert and machine agree or disagree that *n* is properly included in the set of documents classified by C.

	Domain expert classifications	**Machine classifications**	
(1)	Y	Y	(Positive agreement)
(2)	Y	N	(Machine misses)
(3)	N	Y	(Noise)
(4)	N	N	(Negative agreement)

Figure 6. Comparison of Domain Expert and Machine Classifications

It became readily apparent, by examining "noise" documents from (3) that many documents classified as "N" by humans had been so classified under the phase-2 indexing rule:

Assign Y to *n* only if {X} occurs in *n*.

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

where {X} was a set of statute text features sufficient to promote the classification of *n* by C. There were then, in the legacy training data, two ways in which a document could have received an "N" classification: (1) where the statute document did not contain {X}; and (2) where it did but the domain expert refused to apply the classification based on pragmatic considerations related to print.[7]

This meant that under phase-1 rules, which are the only ones machines can meaningfully use, a significant number of documents in the (3) cell of the matrix would have received "Y" classification by humans. This suggested that reinspection of these documents by experts ought to result in the reassignment of documents from (3) to (1). Interviews with domain experts revealed that manual indexers, in classifying training documents, routinely eliminated otherwise appropriate index assignments for the purpose of reducing the size of print indexes. This practice caused training data to contain many examples of text features which ought to have been, but were not good predictors of a particular index feature. The overall effect of this was to mistakenly include them in negative training documents as examples of inappropriate assignment of the index feature. To compensate for this effect, domain experts followed a strategy of limited review of noise in results, looking for certain occurrences of machine-assigned terms that were assigned by humans in the first phase of the indexing process, but eliminated in phase-2 for print purposes.

To this end investigators devised strategies to reevaluate only "N" documents that could not have been paired with machine "N" classifications. The most important of these was the identification by domain experts of what were termed 'sure classifiers'. Sure classifiers are text features, culled from a high-precision list, that promote presumptively valid classifications. For example, among the high-precision features found in the training corpus local collection for 'INSURANCE' were:

Text Feature	Local Document Frequency	Global Document Frequency
agent	916	1128
insurer	3277	4360
Lloyd's	24	24

Figure 7 Selected High-Precision Text Features for the Index Feature *Insurance*

Domain experts were prepared to guarantee that any document containing the string 'insurer' or 'Lloyds' promoted a phase-1 classification, and investigators could therefore conclude that the1083 documents not assigned the index feature 'INSURANCE' by human experts must have been demoted under the discretion to withhold classification available to indexers under phase-2 rules. With the text feature 'agent' however, domain experts could not say presumptively that 'agent' would promote an 'INSURANCE' classification, probably in view of the fact that many other index features might affirmatively classify documents having the text feature 'agents'.

As presumptively valid text feature classifiers were found in the noise (cell) precision and recall scores could be recalculated to reflect the real distribution of noise in the results.

8.3 *Inter-indexer inconsistency.*

The question of the general validity of models generated from training examples also raised questions of how the principle of inter-indexer inconsistency ought to affect results. Inter-indexer inconsistency is well-established in manual indexing (Cooper 1969, Salton 1989., p. 297, Sievert & Andrews 1991, Ellis et al. 1994). The question that remains open is whether inter-indexer inconsistency is *insidious*, i.e., whether or not a permanent staff of expert indexers could be trained in rule-based classifications designed to specifically avoid the inconsistency in question. The investigators hypothesized that inconsistency in indexed collections, generally, resulted from at least one of three conditions which they hoped to strategically avoid in selecting the instant test and training collections. First, inconsistency might result from derivative indexing, where index features were as uncontrolled as the text itself, causing multiple entries to topically similar documents to be created using unrelated and unlinked index entries. Second, investigators assumed that where inconsistency historically arose, it arose among independent indexers who

[7] See discussion in section 3.3.

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

had no pre-coordinated rules for categorizing document content and identifying topics accurately enough to warrant assignment of a particular index feature to a candidate text document; and third, investigators observed that such actual inconsistency as was encountered in the instant test and training corpora often resulted from disagreement over phase-2 and phase-3 assignments, where index features were discarded for the purposes of accommodating a print medium.

9.0 Conclusions

Results so far appear to give limited support to our hypotheses: (1) that statutes are machine-classifiable accordingly as identifiable statistical linguistic stresses wax or wane in associated text documents; (2) that the phenomenon of false noise may be neutralized by the discovery of high-precision classifiers in the text of documents whose affirmative classification was withheld by domain experts for relations relating to print publications; and (3) that inter-indexer inconsistency is not insidious, but may also be neutralized by providing expert classifiers with coordinated rule-based training. Though for technical reasons, we have not yet been able to formally process false noise identification in training documents, informal review in an earlier experiment suggests that elimination of false noise from the training data will improve results significantly. Results so far support the hypothesis that a substantial proportion of index features are highly amenable to automatic assignment, that others can be assigned accurately with some manual intervention, while a few resist automatic assignment, even with considerable manual intervention.

References

Apte, C., Damerau, F., & Weiss, S. (1994). Towards language independent automated learning of text categorization methods. In Croft, W. B. & van Rijsbergen, C.J. (Eds.). *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)* (pp.23-30). Berlin: Springer-Verlag.

Cohen, W. & Singer, Y. (1996). Context-sensitive learning methods for text classification. In Frei, H.-P., Harman, D., Schauble, P., & Wilkinson, R. (Eds.) *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96)* (pp.307-315). Konstanz, Germany: Hartung-Gorre Verlag.

Cleveland , Ana and Donald (1990). Introduction to Abstracting and Indexing. Englewood, CO. Libraries Unlimited.

Cooper, W.S. (1969). Is interindexer consistency a hobgoblin? *American Documentation*, 20, 268-278.

Creecy, R., Masand, B., Smith, S., Waltz, D. (1992). Trading MIPS and memory for knowledge engineering. *Communications of the ACM* .35, 48-64.

Ellis, D., Furner-Hines, J., & Willet, P. (1994). On the measurement of inter-link consistency and retrieval effectiveness in hypertext databases. In Croft, W. B. & van Rijsbergen, C.J. (Eds.). *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)* (pp.51-60). Berlin: Springer-Verlag.

Hayes, P. J. (1992). Intelligent high-volume text processing using shallow, domain-specific techniques. In Jacobs, P. (Ed.) *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval* (pp. 227-241). Hillsdale, N.J.: Lawrence Erlbaum.

Hodges, J., Yie, S., Reighart, R., & Boggess, L. (1996). An automated system that assists in the generation of document indexes. *Natural Language Engineering*, 2, 137-160.

Leung, C.-H. & Kan, W.-K. (1997). A statistical learning approach to automatic indexing of controlled index terms. *Journal of the American Society for Information Science* , 48, 55-66.

ISSN: 2324-9773

Curran, T., & Thompson, P. (1997). Automatic Categorization of Statute Documents. 8th ASIS SIG/CR Classification Research Workshop, 19-30. doi:10.7152/acro.v8i1.12725

Lewis, D. (1992). Representation and learning in information retrieval. Ph.D. Thesis, Computer Science Department, University of Massachusetts, Amherst, Technical Report 91-93.

Lewis, D., Shapire, R., Callan, J., & Papka, R. (1996). Training algorithms for linear text classifiers. In Frei, H.-P., Harman, D., Schauble, P., & Wilkinson, R. (Eds.) *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96)* (pp.307-315). Konstanz, Germany: Hartung-Gorre Verlag.

Lewis, D. & Gale, W. (1994). A Sequential algorithm for training text classifiers. In Croft, W. B. & van Rijsbergen, C.J. (Eds.). *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)* (pp.3-12). Berlin: Springer-Verlag.

Lewis, D. & Ringuette, M. (1994). A comparison of two learning algorithms for text categorization In Information Science Research Institute, University of Nevada, Las Vegas (Ed.). *Proceedings Third Annual Symposium on Document Analysis and Information Retrieval* (pp. 81-93). Las Vegas: University of Nevada, Las Vegas.

Masand, B., Linoff, G., & Waltz, D. (1992). Classifying news stories using memory based reasoning. In Belkin, N., Ingwersen, P., & Pejtersen, M. (Eds.) *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94).* New York: ACM Press, p.59-65.

Moulinier, I. (1997). Feature selection: a Useful preprocessing step. *19th Annual BCS-IRSG Colloquium on IR Research* (pp. 1-11).

Moulinier, I., Raskinis, G., Ganascia, J.-G. (1996). Text categorization: a symbolic approach. In Information Science Research Institute, University of Nevada, Las Vegas (Ed.). *Proceedings Fifth Annual Symposium on Document Analysis and Information Retrieval* (pp. 87-99). Las Vegas: University of Nevada, Las Vegas.

NISO (1993). Guidelines for Indexes and Related Information Retrieval Devices, ANSI/NISO Z39.4-199x; ISSN:1041-5653 , p. 37 NISO Press.

Quinlan, R. (1993). C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann.

Quinlan, R. (1997). http://www.rulequest.com.

Salton, G. (1989). Automatic Text Processing (pp. 281, 286-287). Reading, MA: Addison-Wesley.

Sievert, M. C. & Andrews, M. J. (1991). Indexing consistency in Information Science Abstracts. *Journal of the American Society for Information Science*, 42, 1-6.

van Rijsbergen, C.J. (1979). Information Retrieval 2d. edition (pp. 174-175). London: Butterworths.

ISSN: 2324-9773