A Machine Learning Approach to Identifying Sections in Legal Briefs

Scott Vanderbeck and **Joseph Bockhorst**
Dept. of Elec. Eng. and Computer Science
University of Wisconsin - Milwaukee
P.O. Box 784 , 2200 E. Kenwood Blvd.
Milwaukee, WI 53201-0784

Chad Oldfather
Marquette University Law School
P.O. Box 1881
Milwaukee, WI 53201-1881

Abstract

With an abundance of legal documents now available in electronic format, legal scholars and practitioners are in need of systems able to search and quantify semantic details of these documents. A key challenge facing designers of such systems, however, is that the majority of these documents are natural language streams lacking formal structure or other explicit semantic information. In this research, we describe a two-stage supervised learning approach for automatically identifying section boundaries and types in appellee briefs. Our approach uses learned classifiers in a two-stage process to categorize white-space separated blocks of text. First, we use a binary classifier to predict whether or not a text block is a section header. Next, we classify those blocks predicted to be section headers in the first stage into one of 19 section types. A cross-validation experiment shows our approach has over 90% accuracy on both tasks, and is significantly more accurate than baseline methods.

Introduction

Now that most of the briefs, opinions and other legal documents produced by court systems are routinely encoded electronically and widely available in online databases, there is interest throughout the legal community for computational tools that enable more effective use of these resources. Document retrieval from keyword or Boolean searches are key tasks that have long been a focus of natural language processing (NLP) algorithms for the legal domain. However, the simple whole document word-count representations and document similarity measures that are typically employed for retrieval limits their relevance to a relatively narrow set of tasks. Practicing attorneys and legal academics are finding that the existing suite of tools fall short of meeting their growing and complex information needs.

Consider, for example, Empirical Legal Studies (ELS), a quickly growing area of legal scholarship that aims to apply quantitative, social-science research methods to questions of law. ELS research studies are increasingly likely to have a component that involves computational processing of large collections of legal documents. One example, are studies of the role of ideological factors that assign an ideology value to legal briefs (*e.g.,* conservative or liberal (Evans *et al.* 2006)). One problem that may arise in settings like this that employ a general similarity measure not tailored to the task at hand is that documents are more likely to group by topics, for instance the type of law, than by, say, ideology.

One general technique that has the potential to improve performance on a wide range of ELS and retrieval tasks is to vary the influence of different sections of a document. For example, studies on ideology, may reduce the influence of content in the "Statement of Facts" section while increasing the influence of the "Argument" section. However, although most briefs have similar types of sections, there are no formal standards for easily extracting them. Computational techniques are needed. Toward that end, we describe here a machine learning approach to automatically identifying sections in legal briefs.

Problem Domain

Our focus here is on briefs written for appellate court cases heard by the United States Courts of Appeals. The appeals process begins when one party to a lawsuit, called the appellant, asserts that a trial court's action was defective in one or more ways by filing an appellant brief. The other party (the appellee) responds with an appellee brief, arguing why the trial courts action should stand. In turn, the appeals court provides its ruling in a written opinion. While there is good reason to investigate methods for identifying structure in all three kinds of documents, for simplicity we restrict our focus here to appellee briefs. We conduct our experiment using a set of 30 cases heard by the First Circuit in 2004.

In the federal courts, the Federal Rules of Appellate Procedure require that appellant briefs include certain sections, and that appellees include some corresponding sections while being free to omit others. There is, however, no standard as to section order or how breaks between sections are to be indicated. Moreover, parties often fail to adhere to the requirements of the rules, with the result being that authors exercise considerable discretion in how they structure and format the documents.

Related Work

Many genres of text are associated with particular conventional structures. Automatically determining all of these types of structures for a large discourse is a difficult and unsolved problem (Jurafsky & Martin 2000). Much of the

previous NLP work in the legal domain concerns Information Retrieval (IR) and the computation of simple features such as word frequency (Grover *et al.* 2003).

Additional work has been done in the legal domain with the focus on summarizing documents. Grover *et al.* developed a method for automatically summarizing legal documents from the British legal system. Their method was based on a statistical classifier that categorized sentences in the order that they may be seen as a candidate text excerpt in a summary (Grover *et al.* 2003).

Farzindar and Lapalme (2004) also described a method for summarizing legal documents. As part of their analysis, they performed thematic segmentation on the documents. Finding that more classic method for segmentation (Hearst 1994; Choi 2000) did not provide satisfactory results, they developed a segmentation process based on specific knowledge of their legal documents. For their study groups of adjacent paragraphs were grouped into blocks of text based on the presence of section titles, relative position within the document and linguistic markers.

The classic algorithm for topic segmentation is TextTiling where like sentences and topics are grouped together (Hearst 1997). More general methods for topic segmentation of a document are generally based on the cohesiveness of adjacent sentences. It is possible to build lexical chains that represent the lexical cohesiveness of adjacent sentences in a document based on important content terms, semantically related references, and resolved anaphors (Moens & De Busser 2001). Lexical chains and cohesiveness can then be used to infer the thematic structure of a document.

In contrast to approaches such as these that are based on inferring the relatedness of sentences in section bodies, our approach focuses identifying and catagorizing section headers. These general approaches are complementary as it would be relatively straightforward to construct a combined method that considers both headers and bodies.

Overview

Our analysis begins with a pre-processing step that converts documents to sequences of text blocks, roughly at the paragraph level (see below for details). We next construct feature vector representations for all blocks. Labeled training sets and supervised learning methods are used to induce two kinds of classifiers: one for distinguishing section header blocks from non-header blocks, and one for classifying the section type of headers. Figure 1 shows a flowchart of the processing for classifying a block of text in the test set. Note that although the type of non-header blocks is not predicted directly, after classifying of all blocks in a document the predicted section for a non-header block is given by the type of the nearest preceding section header.

Models and Methods

Dataset

Appellee briefs in our dataset are available as HTML files. The HTML is not well formed or standardized and provides little insight into the structure of the briefs. The HTML elements do not contain attributes, block level elements, id's, classes, *etc.* that may indicate section breaks or section types. Further, document formatting is inconsistent and non-standardized. For example, one author may use italics for section headings, another bold, while yet another uses inline text. Formatting sometimes even varies from section to section within the same document. Thus, we ignore formatting such as italics or bold, and focus our analysis on the word and character sequenece.

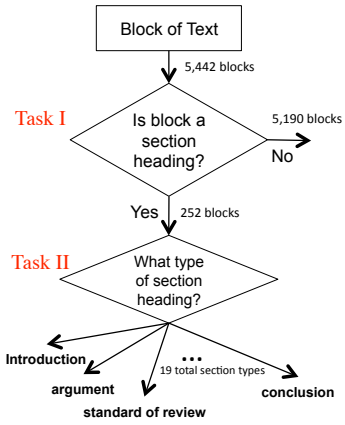

Figure 1: Flowchart of our two stage process for classifying text blocks. The first stage predicts whether or not a block of text is a section header. No further processing is done on blocks classified as non-headers. Blocks classified as headers are passed to the next stage, which predicts the section type. Numbers next to the arrows denote the total number of blocks in our annotated dataset that assort to that point.

Preprocessing was performed on the documents to divide the documents into blocks of text. A block of text is essentially a continuous sequence of text from the original document with a line break immediately before and after. We extract blocks by converting each HTML document to an XML document that recognizes all of the line breaks and white spaces from the original HTML. Examples of document elements that correspond to blocks extracted from the XML include paragraphs, section headings, section sub-headings, footnotes, and table-of-contents entries.

The XML files were manually reviewed and annotated by the author (SV). Each block is assigned two class labels:

1. is_header A binary value indicating whether or not a block is a section heading.
2. section_type A discrete value that for section headers only indicates section type. As we only predict the type of header blocks, the value of "None" is assigned to non-headers. Table 1 shows the section types we identified in our dataset.

Feature Vector Representation

Along with the two class labels, we represent each block of text with a 25 element vector of features values. Ta-

Argument	Notice To Adverse Party	Statement of Parent Companies
Bond	Prayer	And Public Companies
Conclusion	Preliminary Statement	Statement of The Case
Corporate Disclosure Statement	Procedural History	Summary of The Argument
Introduction	Relief Sought	Table of Authorities
Issue Presented For Review	Standard of Review	Table of Contents
Jurisdictional Statement	Statement of Facts	None

Table 1: The 20 section types in our dataset. Each predicted header block is classified as one of the 19 types other than "None."

(a)

Feature Name	**Domain**	**Description**
leadingAsterisk	binary	True if the block begins with an asterisk (*)
leadingNumeral	binary	True if the block begins with an Arabic or Roman numeral (optionally preceded by an asterisk).
endsInPeriod	binary	True if the block ends with a period (.)
endsInNumeral	binary	True if the block ends with an Arabic or Roman numeral.
stringLength	integer	Number of characters in the block.
percentCaps	continuous, in [0,1]	The % of alpha characters that are capitalized.
ellipses	binary	True if the block contains an ellipses (i.e. "...").
contains("argument")	binary	Each of these features is an indicator for a specifc string. The feature contains(s) is true if the block contains a word that begins with the string s and false otherwise.
contains("authori")	binary	
contains("case")	binary	
contains("conclusion")	binary	
contains("contents")	binary	
contains("corporate")	binary	
contains("disclosure")	binary	
contains("fact")	binary	
contains("issue")	binary	
contains("jurisdiction")	binary	
contains("of")	binary	
contains("prayer")	binary	
contains("present")	binary	
contains("review")	binary	
contains("standard")	binary	
contains("statement")	binary	
contains("summary")	binary	
contains("table")	binary	

(b)

leadingAsterisk:	FALSE	contains("of"):	TRUE
endsInPeriod:	FALSE	contains("table":	TRUE
stringLength:	21	contains("contents":	TRUE
percentCaps:	1	(all other string match features):	FALSE
leadingNumeral:	TRUE		
endsInNumeral:	FALSE	is_header:	TRUE
ellipses:	FALSE	section_type:	Table of Contents

Table 2: (a) Features we use to represent blocks of text. (b) An example showing feature and class values for the block of text **"II. TABLE OF CONTENTS"**

ble 2(a) lists the features we use, Table 2(b) shows the feature and class values for the block of text "II. TABLE OF CONTENTS".

The features chosen were engineered through visual inspection of section headings, intuition, and trial and error. Other attributes were considered such as the length and percentage of capital letters of the previous and next blocks of text, however, these did not improve model performance. The group of features named contains(s) are string matching features, which are true if the block of text contains exactly one word that begins with the string s. We construct a string match feature from all words that occur five or more times in the 252 header blocks.

Learning

The task of identifying section headers and the type of section is divided into two steps (Figure 1). The first step classifies a block of text as either a section heading or not a section heading. For this task, supervised machine learning algorithms are used to learn a binary classifier. The second task takes each block of text classified in the first step as a heading and uses a second classifier to predict the specific type of section. Again supervised machine learning is used to learn a classifier, this time with 19 classes. For both tasks, multiple types of classifiers including naive Bayes, logistic regression, decision trees, support vector machines and neural networks were considered.

Evaluation

With the abundance of legal documents available, it is important that they be structured in ways usable by computers (Wynera 2010). We hypothesize the task of structuring our legal documents into relevant sections can be accomplished with a supervised machine learning classifier that first identifies section headers, and then assigns a section type to the header.

To test this hypothesis we have conducted an experiment on 30 appellee briefs from cases heard by the US 1st Circuit in 2004. No effort was made to restrict the cases to a particular area of the law, and indeed a variety of different types of cases is represented in this set. The legal briefs were obtained as HTML files through WestLaw (www.westlaw.com). In the 30 documents, a total of 252 section headers were identified. Note that subsection headers are not included as part of this task as there is very little commonality in authors use of subsections. Additionally, subsections are generally specific to the legal case being addressed, and not the overall document. Of the 252 total section headers, 116 unique strings were identified (not accounting for any difference in formatting or upper / lower case). Manual inspection of the 116 variations revealed that the headers cluster into the 19 different section types listed in Table 2(b). A 20th section type "None" was added to be used as the class label for blocks of text that do not represent section headers.

We conducted a leave-one-case-out cross-validation experiment. That is, in each experiment all blocks from one of our documents was held out of the training set and used as test data to estimate our models' ability to generalize to unseen documents.

For the first task, all blocks of text in the training set are used. For the second task, only training set blocks of text labeled as section headings are used for training. This decision was made because we only wish to use the second classifier to label the section type of true section headers. Also, this approach sidesteps the inconsistency that arises when a block of text is identified as a heading in the first stage, but as section type "None" in the second stage. We may revisit this decision in future work as a "None" prediction in stage two could potentially be used to catch false positives from the first stage. With the current dataset, however, it was found that the number of correctly identified headings being labeled as "None" vs. the correction of false positives was not worth the tradeoff. Therefore, we take the approach described above.

We evaluate models on the first task by the percentage of headings or non-headings correctly classified as well as precision and recall rates where:

$$precision = \frac{\#true\ positives}{\#true\ positives + \#false\ positives}$$

and

$$recall = \frac{\#true\ positives}{\#true\ positives + \#false\ negatives}$$

Note blocks of text that are a section header represent our positive class. Precision and recall are both of particular importance for our first task. Examining our dataset, 95.4% of blocks of text are non-headings. The extreme case of classifying all blocks of text as non-headings would then result in very high overall accuracy and 100% recall rate for non-headings, at the expense of poor precision.

We compare our machine learning approach to a regular expression baseline. The regular expression used for this baseline approach may be summarized as the concatenation of the following list of parts:

1. The beginning of the string
2. An optional asterisk
3. An optional Roman Numeral or Natural Number followed by an optional period and space
4. A list of zero or more all capitalized words
5. The end of the string

Blocks that contain a match to the regular expression are predicted to be headers. This regular expression should correctly identify many section headings as many are entirely capitalized, while excluding false positives such as table of contents entries that are generally followed by a page or section number of some form.

Our second task is then evaluated in two ways. The first is the overall percentage of predicted headings that are assigned the correct section heading type. The second metric is an adjusted metric that does not penalize the second task for errors made in the first task. If the input to the second classification task was a non-heading to begin with, this classifier would inherently fail as it is attempting to determine

the section heading type when no such type actually exists. Therefore, we account for this disparity in our results and also present the number of section heading types predicted correctly divided by actual headings correctly classified by the first task.

A baseline approach is only considered for the first task of identifying whether or not a block of text is a section heading. A baseline approach for the secondary task of assigning a label of one of our 20 classes could be developed through a complicated regular expression or a form of sequential logic, but was not considered in this project. Our most frequent section heading type, "Argument", accounts for 12% of cases. Therefore, that level of accuracy could be achieved by simply always predicting "Argument".

Last, a combined metric is presented where we merged the results from both steps of classification to determine the overall percentage of section headings that are correctly identified and assigned the correct type.

Results

Task 1 - Identifying Section Headings

A total of 5,442 blocks of text were identified in our dataset. Table 3 shows a comparison of the baseline method with our supervised machine learning based approach for the task of identifying if a block of text is a section heading or not. With the exception of naive Bayes (which performed worse), all other classifiers performed similarly.

	Baseline	**Learning Based**
Total Blocks of Text:	5442	5442
Correctly Classified:	5288	5409
Percentage Correct:	97.2%	99.4%

Table 3: Results classifying section headings vs. non section headings

As expected the baseline approach performed very well with 97.2% accuracy. This, represents a small gain over calling all blocks non-headings (95.4%). As we hypothesized, the learning based classifier performed much better with 99.4% accuracy. As seen in the confusion matrix in Table 4, the logistic regression classifier had a similar number of false positives and false negatives. Precision and recall statistics are presented in Table 5. As seen in the table, there is a significant difference in the recall rates of headings (92.1% vs. 61.5%) which is of great importance to the ultimate goal.

	Learning Based		Baseline	
Actual/ Predicted	Heading	Non-Heading	Heading	Non-Heading
Heading	232	20	155	97
Non-Heading	13	5177	57	5133

Table 4: Confusion matrix for Task 1

	Precision	Recall	F-Measure
Learning Based	0.947	0.921	0.934
Baseline	0.731	0.615	0.668

Table 5: Precision and recall of headings for learning based classifier vs. baseline approach

Examining incorrectly classified blocks, the most frequent was "Standard of Review" and accounted for 24% of all errors. Examination of this reveals that the "Standard of Review" is often included as a subsection of the "Argument" section of the brief by many authors, while others choose to make a standalone section. For example, the block of text "1. STANDARD OF REVIEW" was incorrectly classified as a heading in one instance. In this case the author did not use a numbering scheme for the primary section ("Argument" in this case), but numbered the sub-sections on the document confusing our model. Similar errors occurred for the section type "Statement of Facts" and accounted for 12% of all errors.' With additional post processing of the classification, it may be possible to account for these types of errors further increasing model performance.

Task 2 - Predicting Section Type

Table 6 summarizes the result of the secondary classifier that assigns section types to any block of text classified as a heading by the first task. The first task identified 245 blocks of text as headings. Of these, only 18 were assigned an incorrect section heading type for an overall accuracy of 92.7%. However, 13 of these 18 were not actually classes to begin with so the secondary classifier could not have assigned a correct class label. Adjusting for this, 232 blocks of text were correctly identified as headings and of these only 5 were given an incorrect label for an adjusted 97.8% accuracy.

	Count	**Correctly Labeled**	**Percent Correct**
Total Headings Identified	245	227	92.7%
Actual Headings Identified	232	227	97.8%

Table 6: Results of secondary classifier assigning class labels

Combined Accuracy

Combining accuracy from each of the two tasks results in an overall recall rate of 90.1% as seen in Table 7 . Of 252 total labels, 232 were correctly identified as labels. Of those identified, 227 were assigned there correct actual class.

Conclusion

We presented a supervised machine learning approach for structuring legal documents into relevant sections. Our approach is based on two steps. The first step identifies blocks of text that are section headings. In the second step, blocks

Actual Headings	Correctly Identified	Recall Rate	Correct Class	Overall Recall
252	232	92.1%	227	90.1%

Table 7: Combined accuracy for identifying and classifying section headings

of text classified as section headings are then input into a second step to predict section type.

We evaluated our approach with a cross-validation experiment. The first task of identifying section headers using a binary logistic regression classifier was shown to perform with 99.4% accuracy. The secondary task is then used with 92.7% accuracy to determine the type of section one is looking at. The NLP approach provides a 2.2% improvement in accuracy over the baseline regular expression based approach, and more importantly provides a significantly higher recall rate in identifying section headings vs. non section headings.

While it may be possible to create a non-learning based approach (more complex than the baseline approach presented) to perform the given subtask, it has been shown that a machine learning and NLP approach are very well suited for this problem. This paper only researched appellee briefs, but there is ample reason to believe that this approach would provide similar results for appellant briefs, the judges written opinion, and other similar documents.

The significance of our learned models having significantly higher recall rates than baseline models becomes of even greater importance when one considers that approaches would be available to correct or account for false positives (i.e. non-headings classified as headings), however, it would be far more difficult, if even possible, to correct for false negatives (i.e. actual headings classified as non-headings).

While not formally discussed in this paper, it is possible to implement secondary logic to correct for some of the classification errors we encountered. For instance, our most frequent error in the first task was the "Standard of Review. Logic could be implemented as a post processing step that says if a block of text is called a section heading and classified with the section heading type "Standard of Review, but is preceded by the section type "Argument, remove this as a section heading. In our dataset this correction would correct 5 of 7 mistakes made labeling "Standard of Review" and improve accuracy for the first task to 99.5% and 94.6% for the second task.

In addition, allowing the secondary classifier to identify sections that it assigns the class label "None could correct some false positives incorrectly classified as section headings by the first task. In our dataset, 4 such corrections could have been made further improving accuracy. However, if implementing this change one must consider the implications of giving an actual section break heading the section type "None versus the improvement from corrections.

We considered 20 different potential class labels for each section. For specific tasks it may be found that this number can be reduced to even as few as two (i.e. relevant or non-relevant) sections. This could be done as part of the classification or as part of a post process mapping the classifications output by the classifier to a smaller groups of classes for the ultimate task. This may potentially further improve overall performance.

In our approach, the secondary task was treated as individual classifications. It may be possible to treat the secondary classification problem as a Hidden Markov Model or Continuous Random Field. Doing so may improve performance as when an author does include a section in his/her legal briefs, they are generally in a consistent order.

Last, the majority of misclassifications in both tasks appears to be the result of sparse data and infrequently used section headings. While learning curves were not created, it is suspected that additional data could provide the classifier with information about many these sections and improve overall model performance.

With the current model, and the potential for further future improvements, section related information can reliably be identified with supervised machine learning based methods in poorly structured legal documents.

References

Choi, F. Y. Y. 2000. Advances in Domain-Independent Linear Text Segmentation. In *Proc. Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 26–33.

Evans, M. C.; McIntosh, W. V.; Lin, J.; and Cates, C. L. 2006. Recounting the Courts? Applying Automated Content Analysis to Enhance Empirical Legal Research. *SSRN eLibrary*.

Farzindar, A., and Lapalme, G. 2004. Legal text summarization by exploration of the thematic structures and argumentative roles. In *In Text Summarization Branches Out Conference held in conjunction with ACL 2004*, 27–38.

Grover, C.; Hachey, B.; Hughson, I.; and Korycinski, C. 2003. Automatic summarisation of legal documents. In *Proceedings of the 9th international conference on Artificial intelligence and law*, ICAIL '03, 243–251. New York, NY, USA: ACM.

Hearst, M. A. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, 9–16. Stroudsburg, PA, USA: Association for Computational Linguistics.

Hearst, M. A. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics* 23(1):33–64.

Jurafsky, D., and Martin, J. H. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall, 1 edition. neue Auflage kommt im Frhjahr 2008.

Moens, M.-F., and De Busser, R. 2001. Generic topic segmentation of document texts. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, 418–419. New York, NY, USA: ACM.

Wynera, A. 2010. Weaving the legal semantic web with natural language processing. http://blog.law.cornell.edu/voxpop/2010/05/17/weaving-the-legal-semantic-web-with-natural-language-processing.