

Identifying Facts for TCBR

Jason M. Proctor, Ilya Waldstein, Rosina Weber

College of Information Science & Technology, Drexel University
{jp338, imw22, rw37}@drexel.edu

Abstract. This paper explores a method to algorithmically distinguish *case-specific facts* from potentially reusable or adaptable elements of cases in a textual case-based reasoning (TCBR) system. In the legal domain, documents often contain case-specific facts mixed with case-neutral details of law, precedent, conclusions the attorneys reach by applying their interpretation of the law to the case facts, and other aspects of argumentation that attorneys could potentially apply to similar situations. The automated distinction of these two categories, namely *facts* and *other elements*, has the potential to improve quality of automated textual case acquisition. The goal is ultimately to distinguish case problem from solution. To separate fact from other elements, we use an information gain (IG) algorithm to identify words that serve as efficient markers of one or the other. We demonstrate that this technique can successfully distinguish case-specific fact paragraphs from others, and propose future work to overcome some of the limitations of this pilot project.

1 Introduction

Case-based reasoning (CBR) is a problem-solving methodology built on the foundation that if prior problems were solved and recorded, new similar problems can be solved by adapting old solutions [8]. The field of textual case-based reasoning (TCBR) [5] started to attract attention when researchers noticed that, in many domains, useful episodes for CBR were available in textual format. By using representations based on text, cases are able to capture domain knowledge in fields where the reusable experience is based on language [6]. Recent work is incorporating natural language processing (NLP) techniques to support machine learning (e.g. [1]), with an eye towards automated analogical reasoning, but those are still computationally expensive and not yet scalable to large systems.

Cunningham et al. [2] proposed a case representation based on graphs, which improved on the common bag-of-words approach by maintaining word order, and giving higher weight to similarity based on n-grams. Their approach showed promise for graph similarity as a basis for a retrieval algorithm, but it was not able to distinguish case problem from case solution because both of those aspects were intermixed in the source text. This paper aims to refine their representation by separating those aspects, so that similarity of problems and similarity of solutions can be considered independently, rather than together.

We propose an approach that uses information gain (IG), a statistical algorithm to identify predictive elements for classification, to distinguish fact from other language (e.g. interpretations and solutions). In TCBR, these key elements help us separate case problem (facts) from case solution (everything else); by distinguishing them, we expect to improve automated case acquisition of textual cases. This would help TCBR methods (particularly those that do not use NLP) become more accurate. For example, if we integrate this technique into the representation proposed by Cunningham et al. [2], we will be able to create separate graphs for case problem and case solution.

The next section describes the motivation and background for this research. Section 3 describes the algorithms of this pilot study in more detail, and Section 4 shows their efficacy in separating fact from reasoning to establish textual cases. The paper concludes with a discussion of the significance of this work to TCBR and suggestions for further enhancement of these algorithms in Section 5.

2 Motivation and Background

2.1 Representations of Textual Cases

The legal domain is ideal for TCBR because episodes are available in textual format, and given its importance in modern society and heavy reliance on precedent, the legal domain could benefit from automated reasoning. Legal documents typically contain a mix of facts specific to the case and discussion by attorneys about laws, relevant precedent, and other reasoning. CBR systems traditionally represent cases as problem-solution pairs or problem-solution-outcome triples [3]. In legal cases, the facts of a case are the case problem, and that the other elements, e.g. argumentation and conclusions, contain the reusable solution. Automatically distinguishing case facts with little human intervention is an important step for TCBR systems in supporting the attorneys' process by providing relevant background and suggestions. However, to offer a powerful reasoning capability, further work on categorization and processing of the non-factual elements will be needed.

The approach is not intended to be limited to the legal domain; it is meant to be general for interpretive case-based reasoners where case problems typically describe facts, such as the exploration of scholarly publishing. By being able to automatically distinguish facts in source documents where they are intermixed with other elements, a TCBR system would require much less human engineering effort to establish its case base, and could result in improved accuracy by applying NLP and IR techniques, or even radically different representations such as the graph-based forms in [2], to the more focused sections instead of the documents as a whole.

The problem is essentially that of classification: some unit of discourse (a sentence or paragraph) must be classified as either fact or non-fact. In 2004, Wiratunga et al. [15] presented an algorithm for binary classification that identifies important feature words using IG-based metrics, in order to reduce the significant human engineering effort

involved in constructing a textual case base. Their results demonstrated that IG, especially when implemented with a feedback technique (boosted decision stumps) and association rule induction to support the development of generalized features, produces extremely high accuracy (up to 99%) in classifying the content of test documents from four different collections of newsgroup and email messages. In this study, we adapt their basic IG algorithm to select words that signify the presence of fact (case problem) or other elements (case solution) within a paragraph.

2.2 Information Gain

The idea of IG is based on Shannon's theory that the value of a message increases as its likelihood of occurring decreases [11]. In context of vector-based information retrieval, the *message* is usually a word within a vector of known words, and its likelihood of occurring is based on the vocabulary of the database being queried. In a binary vector model with binary classification, the presence or absence of a message contributes some *amount of information* to the ultimate classification of the state of the world. This amount of information can be quantified, and is cumulative; thus, encountering a message means we *gain* information. Messages with high information content are said to be good *signifiers* in a particular classification task.

For example, if a person who knows English sees the word *crayon* in a sentence, the presence of this word (a binary decision) and his/her knowledge that it is an English word (another binary decision: whether it is part of the known vocabulary) contribute to the impression, but are not enough to confirm unequivocally, that the sentence is in English (a binary classification decision). If the sentence continues in French, which also has a word *crayon*, then it is the preponderance of other words that are not English which ultimately allows the reader to mentally label the sentence French. In a context where English and French are both common, the word *crayon* is not a good signifier of either language because it is likely to occur in either; there is no help to the classification process if *crayon* is encountered in text, so there is no information gain. On the other hand, *yellow* and *jaune* have fairly high potential for distinguishing English from French, because they do not frequently occur outside their own languages.

$$IG(X, Y) = \sum_{x \in \{0,1\}} \sum_{y \in \{0,1\}} P(X = x, Y = y) \cdot \log_2 \frac{P(X = x, Y = y)}{P(X = x) \cdot P(Y = y)} \quad (1)$$

The IG score of a message can be calculated using Equation 1 (copied from [15]). $P(X = x)$ indicates the probability that the message is ($x = 1$) or is not ($x = 0$) in any document in the corpus. $P(Y = y)$ indicates the probability that the message is present in documents that are ($y = 1$) or are not ($y = 0$) members of the particular category into which we are attempting classification. $P(X = x, Y = y)$ indicates, in turn, the probability of each of the four possible conjunctions of the above X and Y states.

It is important to note that an IG score is not a probability; it is an indicator of the relative strength of predictive power. IG scores are only meaningful in comparison to each other, and only over the same corpus and categorization: higher scores mean the word is a better signifier of category within the corpus, but there is no distinction whether its presence or absence is the actual marker. In a typical application, the IG score is computed for every word present in the corpus, then that list is sorted and pruned to some subset of highly predictive signifier words.

2.3 Weighted Vector-Based Information Retrieval

To perform the classification, we will also need an algorithm that matches a *query* expressed as a list of words (the output of the IG algorithm) to the documents in the corpus and ranks the results. Salton's cosine-based similarity metric [9] is ideal, because it allows for the possibility of *weighting* the words in the match based on some external importance criterion [14]. The most common weighting method, term frequency times inverse document frequency ($tf*idf$), is closely related to the idea of IG: words that occur frequently in fewer documents should be given more weight. The difference between IG and $tf*idf$ is that IG also has a component to satisfy the classification step, where $tf*idf$ looks only at frequency of occurrence.

A word's normalized term frequency (tf) score is the quotient of the number of times it appears in a document and the highest number of times any word appears in the document. Thus, the most frequently occurring word in a document has $tf = 1$, and as words occur less often, their tf approaches 0. A word's inverse document frequency (idf) is a measure of its distribution throughout the corpus; it equals the logarithm of the number of documents N divided by the number of documents in which the term appears n . If a word appears in every document, $N = n$, the quotient N/n is 1, and the logarithm of 1 is always 0, so that term's $idf = 0$, thus its $tf*idf$ score, the product of tf and idf , must also be 0. If a term appears in only document in the corpus, its $idf = \log N$.

The result of a cosine-based similarity metric is a single number in the range 0 to 1, inclusive, for every document in the corpus. This number can be considered a distance score between the signifier list and the target document: the lower the number, the closer the match, so the more likely it is that the document satisfies the query.

3 Methodology

In the context of this experiment, the messages are words, and the documents are paragraphs within texts instead of the complete texts. Because IG is highly dependent on the inverse of the size of the word list (in all of the probabilities being considered), and is not normalized, IG scores will appear higher in this application than in many other reported uses because of our relatively small corpus.

Our first step is to establish the list of features, sorted by their IG scores. To do this, we apply Equation 1 to a set of training documents that have been manually separated into fact and non-fact. Once the list of highly predictive signifiers (words with high information gain) is established, it must be applied to each document in the remainder of the corpus in order to classify it. This is exactly parallel to many common models of information retrieval, where the predictive signifiers are the query terms, and properly classified documents are those that match the query.

In order to use term weights to improve the cosine similarity metric, we must calculate them for the query vector (the signifiers). The words' IG scores contain an inherent ambiguity that makes them unsuitable for use as weights: we do not know if they are signifiers of fact or non-fact elements. To overcome this, the *tf* part of the *tf*idf* weight, which favors a term's repeated use, is calculated following Salton and Buckley's modification to the basic *tf*idf* formula [10] only on the paragraphs that have been classified as non-fact. Note that we could have chosen to calculate *tf*idf* scores for the words in the fact paragraphs instead, but some preliminary analysis revealed that the non-fact words were more consistent across the data. In a binary classification such as this, the results should be similar.

In addition to the weights for the signifiers that make up the query vector, the *tf*idf*-weighted cosine algorithm for selecting paragraphs also requires a *tf*idf* score for each word in each paragraph. By using *tf*idf* on both the signifier list and the words in the target paragraphs, we guarantee signifier words that are considered important (i.e. they have high *tf*idf* scores) are given more influence over the classification decision when they are also considered important in their native paragraph.

Different applications use the similarity score differently; because ours is ultimately a binary decision, there must be a threshold established before it can be applied to test data. Since the classification is based on a probabilistic aggregate of training data, the trained algorithm can be applied to these known documents, and the results can be examined in order to determine a suitable threshold.

4 Evaluation

Our hypothesis is that an IG-based algorithm can help distinguish fact from other elements in source texts for cases. Because classification is effectively a retrieval task, we use precision and recall as measures of quality. Precision is defined as the ratio of correctly retrieved units (i.e. paragraphs labeled as non-fact by both the computer and the humans) to the total number of retrieved units (i.e. all paragraphs labeled as non-fact regardless of correctness). Recall is the ratio of correctly retrieved units compared to the actual number of units that should have been retrieved (i.e. the number of human-classified non-fact paragraphs).

The data set consists of 26 claims summary documents from an insurance company's law firm, also used in the Cunningham et al. study [2]. The documents describe the facts

of personal injury lawsuits, from the point of view of the defendants’ attorneys. They consist mostly of summaries of the depositions of participants and whatever medical records were available at the time of writing, written by the attorneys as an intermediate step in resolving the case. Documents also include an estimated settlement and some of the attorneys’ interpretation of the law and relevant precedent, and sometimes internal budget information.

In each document, proper names and addresses were manually replaced by role-identifying placeholders (e.g. “DefendantEmployee1”) to protect the identities of the people involved, similar to role replacement described in [1]. We tested the hypothesis by training the IG classifier on 6 of the documents (23%), randomly selected, then applying the trained classifier to the remaining 20 documents. A more detailed investigation should employ a technique leave one out cross-validation, but this will require significant human preprocessing effort.

4.1 Training

The paragraphs of training documents were classified as fact or non-fact by three people fluent in English, but not familiar with legal terms. Fact paragraphs are those that are specific to the case at hand, describing the events that led to the personal injury lawsuit; non-fact paragraphs have content that may depend on fact, but could potentially be reused in another case with similar facts. The training program used majority opinion—a paragraph was considered fact if at least two of the three had classified it as such. Overall,

Table 1: Signifier Stems

stem	IG score	<i>tf*idfweight</i>
LIABIL	1.002	1.599
OW	1.000	2.480
LIABL	1.000	2.230
DISCOV	1.000	2.346
REASON	1.000	2.020
EXIST	0.959	2.339
CREAT	0.959	2.233
CONDUCT	0.959	2.069
EVALU	0.937	2.144
FACT	0.918	1.701
OFFER	0.918	2.230
OPINION	0.918	2.069
AWAR	0.918	1.807
VENU	0.883	2.230
CIB	0.883	2.002
LOW	0.883	1.462
DANGER	0.877	2.082
EXTENT	0.877	2.141

there were 76 non-fact paragraphs (17.7% of total paragraphs, average length 45.3 words), and 354 fact paragraphs (average length 55.7 words) in the training set.

Stop words were removed from each paragraph, then the remaining words were stemmed using Porter’s algorithm [7] and converted to be all upper case letters. The non-fact paragraphs had 572 distinct word stems, with the most common being “PLAINTIFF”, “CASE”, “CONDIT”, “LIABIL”, and “ACCID”. The fact paragraphs, which were much more varied because they describe the stories of each case based on depositions and medical records, had 1528 distinct stems; the most common were “PLAINTIFF”, “ACCID”, “REPORT”, “PHYSICIAN”, and “PAIN”.

The complete stem lists were then fed to the IG algorithm, which produced a list of stems sorted by their ability to distinguish fact

paragraphs, as described in Section 2.2 above, and Salton and Buckley’s modified $tf*idf$ weight [10]. Table 1 shows the 18 stems with the highest IG scores and their $tf*idf$ weights. Note that the stems such as “PLAINTIFF” and “ACCID” which appear at the top of both the fact and non-fact word frequency lists are not among the best signifiers; in fact, because of their ubiquity in the documents, they are among the worst.

Wiratunga et al. recommend selecting a maximum number of features to use in the classification process [15]. Given the relatively small size of our corpus, however, we found that terms often have the same IG score, and should therefore be considered equally useful as signifiers, so instead of a simple count we selected four possible IG threshold scores: 0.9 (13 stems), 0.7 (18 stems, shown in Table 1), 0.6 (52 stems), and 0.5 (99 stems). 497 of the 1634 total stems (30.4%) had nonzero IG scores.

The lists of signifier stems were provided as a keyword vector to the classifier algorithm along with the complete (i.e. unseparated) training documents, in order to find an appropriate threshold for the cosine and $tf*idf$ -weighted cosine similarity metrics. The classifier program separated the 6 documents into fact and non-fact, and the resulting files were compared with the human assessments used for determining IG scores. We calculated precision and recall for the range of possible thresholds (at intervals of 0.005) across all 6 documents.

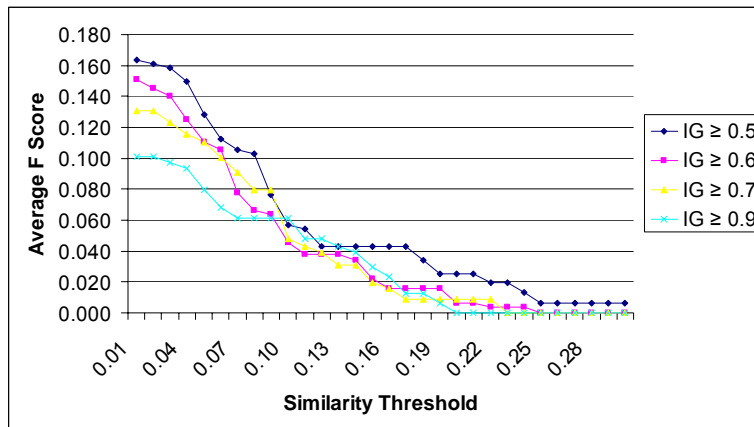


Figure 1: Average F Scores at Different Similarity Thresholds

Figure 1 shows the 6-document average F score (the harmonic mean of precision and recall, suggested by Shaw et al. [12] as a unified measure of retrieval quality) at different similarity thresholds for $tf*idf$ -weighted cosine applied to the signifier words from all four tested IG thresholds. Lower IG thresholds result in longer lists of signifier stems, which therefore produce more precise and more accurate results.

At an IG threshold of 0.5, the steepest decline in quality begins when the classifier requires a similarity score greater than 0.08. This suggests a natural threshold point which can be used by the classifier in labeling the paragraphs from the test documents. At these

values, average precision across the 6 training documents was 6.3%, average recall was 34.1%, and the average F score was 0.103.

4.2 Results

We applied the technique to the 20 test documents not used in training to demonstrate that this technique can successfully distinguish case-specific fact paragraphs from others. As in the training phase above, the classifier used the stem list generated by the training process at the selected IG threshold of 0.5 and similarity threshold of 0.08 to label the paragraphs in the 20 documents. The output of the classifier is a list of paragraphs classified as non-fact; the humans checked this list to determine whether they agreed with the category assignment (using the same criteria as the preprocessing step in developing the training set—whether the contents could potentially be reused). Because fully classifying all 20 documents to get exact counts for recall would be overly time-consuming, we use the human-derived ratio (17.7%) applied to the total number of paragraphs as an approximation of how many non-fact paragraphs there should be in each document.

Table 2 shows averages and standard deviations across the 20 test documents. The first three rows show the results of the classifier program: the number of paragraphs it labeled as non-fact, the number of errors in those classifications as determined by the humans, and the resulting precision. Next is the *estimated* number of non-fact paragraphs that theoretically exist in the documents, based on the 17.7% ratio the humans found in the training set. This is followed by estimated recall and F score based on the estimated number of unselected non-fact paragraphs.

Table 2: Summary of Results

	Average	St. Dev
Number of paragraphs classified as non-fact	4.2	2.6
Number of errors in classification	2.1	1.5
Precision (exact)	50.6%	23.6%
Estimated number of non-fact paragraphs	13.0	6.8
Recall (estimated)	19.4%	14.8%
F score (estimated)	0.265	0.174

4.3 Discussion

Of the 1214 paragraphs in the test documents, 84 (6.9%) were classified as non-fact, which is significantly lower than the humans' percentage of non-fact paragraphs found in the training documents (17.7%). The errors are consistent with those found in any situation where a statistical technique is applied to a small sample: deviation from expectations in a single document is enough to skew the results noticeably. A larger

training set could also improve the list of signifier stems by creating a wider distribution of IG scores and filtering out more of the words that are less common in fact paragraphs.

If the training set's average of 17.7% non-fact paragraphs holds for the entire corpus, then randomly selecting paragraphs should produce a precision of 17.7%. Clearly, the classifier shows reasonable success even with 50.6% precision, although there is still much room for improvement.

5 Conclusions and Future Work

We have applied an IG-based algorithm to classify text at the paragraph level as either fact or non-fact. While additional advancement is necessary before this pilot project can become a practical part of automatic case acquisition in a TCBR system, the relatively high precision and low recall suggest that the scope of that problem is reduced primarily to one of identifying false negatives, perhaps based on some other similarity metric that can include the IG-identified argumentation as part of its input.

The obvious next step is to further explore the quality of evaluation by committing the human effort necessary to classify every paragraph in every document, for a full cross-validation. Classification problems other than fact versus non-fact would contribute to our understanding of the strengths of this technique. More complete classification data about the corpus will also enable us to calculate exact recall scores, as well as explore alternate similarity metrics and threshold values. We can then integrate this technique with the graph representation proposed by Cunningham et al. [2] to test its impact on the quality of retrieval.

Future work could include the addition of the boosted decision stump enhancement to IG suggested by Wiratunga et al. [15], which could improve the ability of the software to identify the most important, least correlated words that indicate fact or argumentation. Mathematically, such a step would require larger training and testing sets to produce meaningful results. In general, the human engineering effort that goes into establishing a test set is the limiting factor in exploring the algorithmic categorization of text. Additionally, a semantic pattern technique such as that described by Weber et al. [13], which includes synonyms and markers that consist of multiple words (which are not necessarily adjacent), would likely catch some of the logical constructs that this single-word analysis misses.

As the classification step becomes reasonably reliable, and tested across additional domains, then machine learning research could be focused on the analysis of the reusable and adaptable elements of each case. Recent work has moved in this direction, incorporating language processing and other techniques, but there is still room for improvement [4]. A technique such as that suggested in [1] could then facilitate retrieval based on the potential to reuse or adapt known solutions instead of surface problem features.

Acknowledgements

The authors would like to thank Caleb Fowler for his suggestions on this work, as well as Amit Deshpande and Bradley Breneisen for help managing the case files.

References

- [1] Brüninghaus, S. and Ashley, K. D., "Reasoning with Textual Cases," in *Proceedings of the Sixth International Conference on Case-Based Reasoning (ICCBR 2005)*; LNCS 3620, H. Muñoz-Avila and F. Ricci, Eds. Berlin: Springer, 2005.
- [2] Cunningham, C., Weber, R., Proctor, J. M., Fowler, C., and Murphy, M., "Investigating Graphs in Textual Case-Based Reasoning," in *Advances in Case-Based Reasoning: 7th European Conference (ECCBR 2004)*; LNAI 3155, P. Funk and P. A. González Calero, Eds. Berlin: Springer, 2004, pp. 573–586.
- [3] Kolodner, J., *Case-Based Reasoning*. San Francisco CA: Morgan Kaufmann, 1993.
- [4] Lamontagne, L. and Lapalme, G., "Textual Reuse for Email Response," in *Advances in Case-Based Reasoning: 7th European Conference, ECCBR 2004*; LNAI 3155, P. Funk and P. A. González Calero, Eds. Berlin: Springer, 2004, pp. 242–256.
- [5] Lenz, M. and Ashley, K. D., "Textual Case-Based Reasoning: Papers from the AAAI-98 Workshop (Technical Report WS-98-12)." Menlo Park CA: AAAI Press, 1998.
- [6] Lenz, M., Hübner, A., and Kunze, M., "Textual CBR," in *Case-Based Reasoning Technology - From foundations to Applications (LNAI 1400)*, M. Lenz, B. Bartsch-Spörl, B. H.-D., and S. Wess, Eds.: Springer Verlag, 1998, pp. 115–137.
- [7] Porter, M., "An Algorithm for Suffix Stripping," *Program*, vol. 14, pp. 130–137, 1980.
- [8] Riesbeck, C. K. and Schank, R. C., *Inside Case-Based Reasoning*. Mahwah NJ: Lawrence Erlbaum Associates, 1989.
- [9] Salton, G., "The SMART Retrieval System: Experiments in Automatic Document Processing." Englewood Cliffs NJ: Prentice Hall, 1971.
- [10] Salton, G. and Buckley, C., "Term-Weighting Approaches in Automatic Retrieval," *Information Processing & Management*, vol. 24, pp. 513–523, 1988.
- [11] Shannon, C. E. and Weaver, W., *The Mathematical Theory of Communication*. Urbana IL: University of Illinois Press, 1949.
- [12] Shaw, W. M., Jr., Burgin, R., and Howell, P., "Performance Standards and Evaluations in IR Test Collections: Cluster-Based Retrieval Models," *Information Processing & Management*, vol. 33, pp. 1–14, 1997.
- [13] Weber, R., Waldstein, I., Deshpande, A., and Proctor, J. M., "Integrated Approach to Detect Inconspicuous Content," in *Professional Knowledge Management*, K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, and T. Roth-Berghofer, Eds. Berlin: Springer (in press), 2005.
- [14] Wilson, D. C. and Bradshaw, S., "CBR Textuality," *Expert Update*, vol. 3, pp. 28–37, 2000.
- [15] Wiratunga, N., Koychev, I., and Massie, S., "Feature Selection and Generalisation for Retrieval of Textual Cases," in *Advances in Case-Based Reasoning: 7th European Conference (ECCBR 2004)*; LNAI 3155, P. Funk and P. A. González Calero, Eds. Berlin: Springer, 2004, pp. 806–820.