Integrated Approach to Detect Inconspicuous Contents*

Rosina Weber, Ilya Waldstein, Amit Deshpande, Jason M. Proctor

College of Information Science & Technology, Drexel University
3141 Chestnut Street, Philadelphia, PA 19104 USA
{rw37,imw22,asd28,jp338}@drexel.edu

Abstract. This paper describes an integrated approach for detecting inconspicuous contents in text. Inconspicuous contents can be an opinion or goal that may be disguised in some way to mislead automated methods but keeps a clear message for humans (e.g., terrorist sites). Our methodology hypothesizes that patterns that convey inconspicuous contents can be extracted, represented, generalized, and matched in unknown text. The proposed approach is meant to complement data-intensive methods (e.g. clustering). Data-intensive methods are fast but are susceptible to variations in frequency, do not discern meaning, and require a large corpus for training. Our approach relies on manual engineering for natural language interpretation and pattern extraction using no more than ten examples, but is sufficiently fast to complement a real-time application.

1 Introduction

The approach we introduce in this paper integrates intelligent methods and manual engineering to collect, store, and reuse knowledge for the timely identification of inconspicuous contents. We refer to content as inconspicuous if its message is easily recognized by humans but not as easily recognized by automated methods.

Terror-related [1] web pages are an example of inconspicuous contents. Their authors have a focused audience they need to reach, but they do not wish to be recognized by others outside that targeted audience. Email spam [2] is another example of inconspicuous contents. It is not uncommon to see keyword lists inserted in email spam messages. These are attempts to disguise their contents, preventing these email messages from being automatically detected by methods that can delete them. Sentiment can also be classified as inconspicuous contents. The task of sentiment analysis [3] aims at automatically recognizing opposing views, e.g. when a customer reviews a product showing satisfaction or dissatisfaction.

The detection of terror-related web pages with a data mining approach was proposed in [1]. Their method can be viewed as data-intensive because it relies on clustering training data to learn a model of a *typical terrorist behavior*. Their studies demonstrated the proposed method reaches almost 90% accuracy when detecting terror-related pages. Similar to clustering, the detection of email spam also relies on training large amounts of training instances, such as the method in [2]. Sentiments are detected by Yi et al. [3] using natural language processing (NLP) techniques. NLP is expensive in terms of complexity and time and its accuracy, although high, is no match for humans.

In Klaus-Dieter Althoff, Andreas Dengel, Ralph Bergmann, Markus Nick and Thomas Roth-Berghofer (eds.): Professional Knowledge Management. Springer, Berlin.

Our goal is to create a method that can improve the current accuracy of detection of inconspicuous contents. Given the high accuracy obtained by data-intensive methods (i.e. around 90% [1]), we want to complement such methods and contribute with an increase in their accuracy. For this purpose, we assume that lack of meaning is one of the reasons limiting accuracy of data-intensive methods. Another reason we consider is that they rely on generalizations that are usually detrimental to their specificity. In addition, in the context of inconspicuous contents, authors may try to disguise the contents by inserting keywords or excerpts from extraneous text. Therefore, we also take into account that data-intensive methods are susceptible to artificial variations of word frequencies.

The input of our approach is the output of a data-intensive method such as [1]. Their clustering method succeeds in classifying around 90% of the documents. Thus, the remaining 10% of the documents that are undetected become the input to our approach. These undetected documents are likely to contain some surface features that are relevant to the targeted topic, but these words or features are not enough to assign these documents a clear classification so the documents fall below that method's threshold. Therefore, we envision our detection task as one that requires a more specific test. Our assumption is that there are different purposes or perspectives a document may underlie if it is an instance of a given topic. Our purpose is then to detect whether these documents are instances of one of a set of potential perspectives within a topic. If a document carries a perspective of a topic, then it is highly likely it is an instance of that topic. For example, if we assume that terrorist web pages may have one of many perspectives such as organizing a meeting, a document that is around the classification threshold of terrorist pages that also has elements of organizing a meeting is more likely to be an instance of terrorism than not.

The problem we face has a number of characteristics we must take into account. First, we may not have a training corpus available for training. We may have a few examples but we may also need to rely on humans to assume how different angles of a given domain may be expressed. Second, if we want to incorporate our approach into a data-intensive method for real-time detection, the execution time has to be very fast. Detection speed is crucial in domains such as terrorism: detecting terrorist contents on the web is only useful if done in real time so that authorities can be notified to take necessary measures.

Given the definition of the problem and its characteristics, we propose an approach to detect inconspicuous contents that relies on a highly simplified representation formalism for patterns. The proposed approach consists of two steps. A manual engineering step that targets a very small set of training instances, where humans perform the complex understanding offline, extract semantic patterns, and represent them in a simplified representation formalism. In the second step, the patterns are used in real-time detection.

As a complementary method, it takes advantage of the preliminary screening produced by the data-intensive method and only processes the text that ranks below the threshold. Therefore, our classification task aims at detecting when a given text presents a specific perspective or *view* of a topic, and not the topic. Having previous knowledge about the domain of the text to be classified allows us to make some assumptions about semantics in the incoming texts. The essence of the approach is that humans can identify when a combination of words in a given topic indicates one or another view.

The next section highlights related work. Section 3 presents our proposed approach. Section 4 describes our preliminary studies. Conclusions and future work are discussed in Section 5.

2 Related Work

The particular data-intensive application for web detection that motivated our research is Elovici et al. [1]. Their method relies on modeling terrorist behavior based on the particular characteristics of this problem. The distribution of terrorists on the web is neither stable nor balanced. The cost of missing one terrorist is higher than suspecting many legal users [ibid.]. Authors [ibid.] showed their results in a ROC chart [4] where the rate for true positives reached 93% when false positives were at 11.7%, with classification accuracy at 88.9%. These results are hard to improve. Nevertheless, we wanted to develop a method to account for the texts that may fail to classify in their method.

The problem we address is more specific than a topical classification. We already know the domain of the text, what is left to determine is which specific view a text conveys. A view is usually defined by the author's intention. Intent classification for email messages as done in [2] focuses on the goals of the authors of the text. As in [2], our goal is to identify the illocutionary speech act that may trigger some effect on the reader [5]. We are not considering the actual impact as in the modeling of perlocutionary [ibid.] speech acts.

Computational linguistics methods take into account the rhetorical and illocutionary structure of documents. Marcu [6] uses cue phrases to locate and classify rhetorical relations. Branting and Lester [7] represented the rhetorical and illocutionary structure to reuse documents with similar intentional structure. These structures can reveal the intensions of the document's author, which are not necessarily visible in the document's surface text. Consequently, the identification of these structures facilitates interpretation of documents. Cohen et al. [2] also rely on the notion of this analysis to identify illocutionary points of speech acts for classifying email according to intent. The rhetorical structure of a document has also been used for Information Extraction. Template mining [8] is a form of information extraction that makes use of the illocutionary structure of textual documents to extract information from text without using natural language processing.

The patterns we propose here are related to the *factors* concept [9][10]. Factors are portions of text that support one or another argument, which relate to our patterns that are meant to support a view. However, it is much easier to represent an opinion than a factor supporting a legal argument. Different formalisms to represent text excerpts that are evidences of factors were studied in [11]. They have found that using bag-of-words is an adequate representation for factor assignment [ibid.]. However, when they incorporated reasoning, they found their representation formalism called Propositional Patterns (ProPs) is more effective [ibid.]. Bag-of-words are easy to build but they include all words without keeping any reference to ordering or meaning. ProPs are sophisticated representation formalisms that keep ordering, semantics and syntax; but require use of NLP for their construction. In our problem, we wanted to combine sophistication of preserving meaning with simplicity of detection.

Sentiment analysis is performed in [3] with a feature extraction method that relies on NLP. Their resulting accuracy is over 90%. They also use human analysis for validating their patterns. Because their purpose is increasing automation and accuracy, they will explore the use of NLP even further. They are not concerned with real time detection.

Yi et al. [3] used Wordnet to generalize words in a pattern through synonyms. Cohen et al. [2] used ontologies to identify speech acts. Brüninghaus and Ashley [10] used a legal thesaurus for generalizing rules. For generalizing our approach, we rely on humans to define *similars* – alternating words that can be exchanged without altering the validity of a pattern. Although it is theoretically possible to represent similars in highly specialized ontologies of intentional entities [12], given the small number of training instances, it is faster, easier, and maybe less prone to error having humans finding similars.

3 Approach

Figure 1 shows the problem we address with our proposed approach. We are assuming to complement a data-intensive method that may reach accuracy levels around 90% in text classification. Therefore, the input to our module is roughly about 10% of the documents for which the previous method cannot provide a correct classification. In practice, the documents input in our module are the ones whose classifications fall below a predefined threshold of classification.

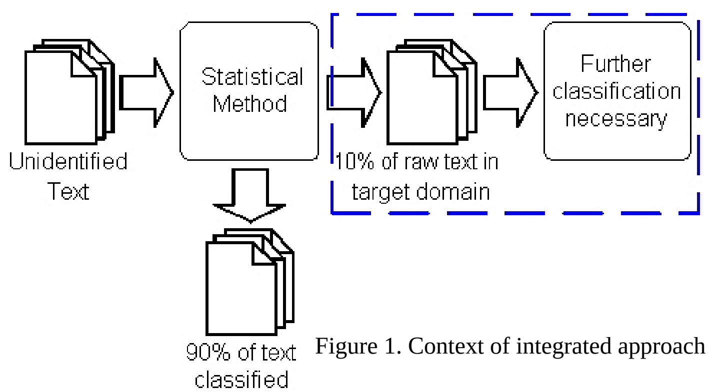

Figure 1. Context of integrated approach

Figures 2 and 3 show the major steps in our approach, which are described in this section. The approach is based on the hypothesis that there is a combination of words that should appear in a text excerpt to characterize it as a view to be understood by human readers. In the manual engineering step, humans extract view structures that represent the target view with patterns and similars, which are used in the detection step.

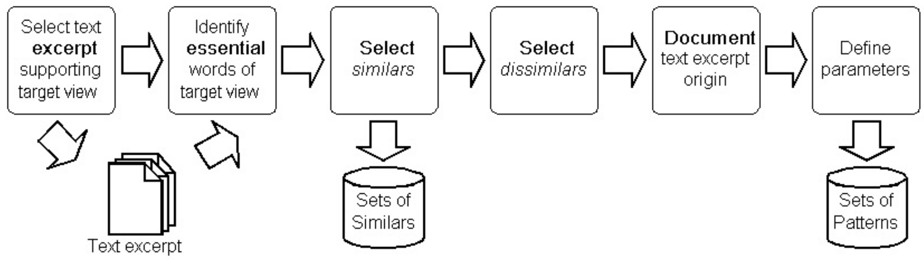

Figure 2. Steps of the Manual Engineering step

3.1 Manual Engineering

The manual engineering step consists of humans interpreting samples of text and extracting and representing patterns that characterize views. A view is described as a set of patterns. A new text is classified as an instance of a view when it contains a minimum number of patterns that characterize such a view.

Table 1. Text and corresponding pattern

Original text	Text excerpt pattern is based on: "And like many other ads for the diet industry (and yes Weight Loss **Surgery** is another diet only it forces the individual to diet by **permanently damaging** their digestive system), the long term results of this **surgery** are seldom if ever, even mentioned.[1]"

Pattern	Star word, W1 = DEFORMATION
	# Words before = 15
	# Words after = 15
	W2 = PROCEDURE
	W3 = PERMANENT

Pattern Extraction. The pattern extraction step requires as input the definition of views and a small set of examples. If examples are not available, engineers need to define patterns by creating examples of excerpts that they recognize as containing the target view. First, knowledge engineers select excerpts that convey ideas supporting the target view. Second, is the identification of words in each excerpt that are essential to the view – representative words, that is, if you removed these words, the view would no longer be present. Third, is the selection of alternative words that could replace these words not affecting the view – these are *similars*. Fourth, selection of words that would invalidate the view – these are *dissimilars*. Fifth, numbering the excerpt and the set of similars to store the location and title of original text where the excerpt came from for future verification of detection efficacy. At the end of this step, all the steps are revised for confirmation of results. The final task is to define the parameters, of the pattern representation (to be detailed next). The label is the word with which all the occurrences in the set of similars will be replaced to create a *canonical* version of the text. Next, the patterns and sets of similars are revised by replacing words in patterns with words in the sets of dissimilars. This step

[1] From http://gastricbypass.netfirms.com/wlssell.htm

is for verification of the patterns. Table 1 shows an example of an excerpt and the pattern extracted from it.

It is important to note that although it is desirable to reduce human judgment from the methodology, the selection of similars is not done exclusively from the training examples. Common sense reasoning and background knowledge is used to select similars and can be used to create additional patterns if it is believed that they can represent a view.

Pattern Representation. Patterns are represented through a list of words and a range where all these words should occur. The representation for the patterns consists of the star word (w_1) and the remaining words (w_2, w_3, ..., w_n). The number of words defined in a pattern (n) is the sum of the star word plus the number of remaining words. The star word is the reference for the pattern's range. Each pattern has a range that is defined with respect to the star word. The range has two directions, before (r_1) and after (r_2) the star word. The representation can be presented as: Pattern #, n, w_1, w_2, w_3,..., w_n, r_1, r_2. For example, Pattern 19 discussed above will be represented as follows, *" 19 3 deformation procedure permanent 15 15."*

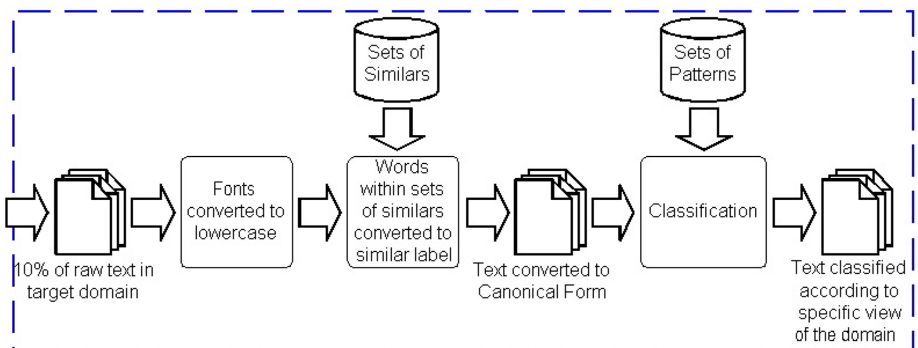

Figure 3. The Detection step

3.2 Detection

Detection consists of pre-processing and pattern matching. In the preprocessing, all fonts are converted into lower case. Then, all words within the sets of similars are converted into the label of the similar, i.e. creating a canonical text. To match a pattern, all words in the pattern must be found within the pattern's range. First, the star word is found, and then the pattern's range is delimited. When the program reaches the end of the text without finding a star word, there is no need to evaluate other patterns that contain this word. The detection returns which patterns are matched in each text. Table 2 shows pseudocode for pattern detection. The study we discuss next will shed some light on the number of patterns required to match in a text for classification.

When implementing the proposed approach, it is important to note that the role of the patterns is to generalize from specific instances found in the small training set. It is a generalization that does not follow induction, since it generalizes based on human perception and not based on a large sample of training examples. The guidelines for the conception of sets of similars is very unrestrictive. Therefore, it is not only prone to error but also possible that the freedom given to human engineers result in patterns that are not able to detect their own excerpt that originated the pattern. Therefore, it is necessary that, after patterns and similar sets are finished, one verification round of tests confirm that patterns are actually able to detect the texts that originated them. This is the reason why we recommend the recording of every source excerpt during pattern extraction.

Table 2. Pseudocode for pattern detection

```
{Read identified patterns from the file patterns.txt and store in an array}
strPatterns := Array of pattern strings
index := 1
Loop until end of file
        pattern := Read one word
        strPatterns[index] := word
        index ++
Close pattern.txt
{Read identified similars from the file similars.txt and store in two arrays}
strMainWord := Array of strings to be replaced
strReplaceWord := Array of replaced strings
index := 1
Loop until end of file
        word := Read first word
        replace_word := Read next word
        strMainWord [index] := word
        strReplaceWord[index] := replace_word
        index ++
Close similars.txt
{Read files for finding patterns}
FOR nFile =1 to NUMBER_OF_FILES
        {Read the contents of the file in an array}
        strBuffer := file contents
        {Read words from strBuffer and store in an array}
        strWords := words (strBuffer)
        {Find and replace the similars}
        IF strWords = strMainWord THEN
                strWords := strReplaceWord
        END IF
        FOR nPattern =1 to NUMBER_PATTERNS
                bFound := Find_Patterns
                IF bFound = 1 THEN
                        DISPLAY "Pattern Found"
                END IF
Function: Find_Patterns
        Input: text as array of String
                patterns as array of String
                before as Int
                after as Int
        Output: Bool
        WHILE end of text DO
                IF text = pattern THEN
                        bFound := 1
                END IF
        END WHILE
```

4 Evaluating Detection of Views about Cosmetic Procedures

In this section, we describe an implementation and an evaluation of the introduced approach in the domain of cosmetic procedures. This domain is suitable to evaluate the approach because among several texts on the topic, we found texts conveying opinions against cosmetic procedures – one view – and others in favor – the opposing view. The existence of opposing views is useful for testing the approach.

In order to guarantee consistency among knowledge engineers, we started with the definition of the view. Our purpose was to detect one view only, the view against cosmetic procedures. The overall goal of the approach is to detect text on the target view that could have an impact on a reader. In the context of negative propaganda against cosmetic procedures, we considered whether the text conveyed an idea that could potentially cause a person to change her or his mind about undergoing a cosmetic procedure. For example, the statements "*The death of Olivia Goldsmith, a well known writer aged 54 yrs, from a heart attack while undergoing a neck lift operation, illustrates its potential dangers*[2]." has the potential to change a readers mind because it implies extremely high risk. The statement "*only vain and insecure people get a face lift*" has the potential to change a readers mind because it attacks the candidate. On the other hand, statements such as "*the media is providing too much favorable coverage for plastic surgery*" do not imply anything about the candidate (reader) or the procedure and thereby are not considered as supportive of a negative view.

Table 3. Three patterns extracted from texts with views against cosmetic procedures

Pattern #	n	w_1	w_2	w_3	r_1	r_2
3	3	complications	deformation	procedure	10	10
16	2	death	procedure	-	10	10
19	3	deformation	procedure	permanent	15	15

4.1 Methodology

We selected eight positive training texts from the web to extract patterns. We used the manual engineering step described in Subsection 3.1 to extract thirty-one patterns. Table 3 shows three patterns we extracted from different texts. Table 4 shows sets of similars selected for pattern 19 in Table 3. One can notice that synonyms are sometimes used, but there is no relation to sentence structure, and the relation to meaning is loose. The only rule is that the word, when combined with other words in the pattern, conveys an idea supporting the target view. These similars were selected from the texts and based on human judgment.

Dataset. The entire test dataset has fifteen positive and fifteen negative instances, a total of thirty texts. We randomly selected three subsets with ten texts each to submit ten texts at a time and observe the average time for classifying ten texts.

Hypothesis. The hypothesis we want to evaluate is whether it is possible to classify unknown texts as offering a particular view of a previously known domain by using

[2] http://menshealth.about.com/cs/surgery/a/cosmetic.htm

patterns extracted manually from eight examples. For this purpose, we use three metrics (also used in [1]) namely, *true positives* (TP), *false positives* (FP), and *accuracy* (Ac). True positives measure the proportion of correctly classified texts that are positive instances of the target view. False positives indicate the proportion of texts that are falsely classified as instances of the target view. Accuracy measures the proportion of correct classifications out of the total number of texts classified.

Table 4. Sets of similars

Set label	Elements
permanent	eternal lifelong life-long eternally permanently
procedure	surgery lasik procedures surgeries implant operation augmentation liposuction breast botox implants lipoplasty bariatric facelift
deformation	deformity pain disfigurement limitation handicap disaster damage injured suffering damaged damaging injury injuries wounded wounds deformations deformation aberrated abnormality defects aberration vision-robbing debilitating harmful traumatic excruciating intensely painful agonizing extreme torture torturing dangerous hazardous alarming robbing serious life threatening lifethreatening life-threatening extreme radical unnecessary radical drastic seriously unwanted unneeded

Results. The results obtained from sets A, B and C, and their averages are presented in Table 5. The numbers of patterns are the minimum required to match in a text so it will receive a positive classification. The best results were obtained with one pattern, with average rates TP = 0.7, FP = 0.2, and Ac = 0.6 (shaded in Table 5). Overall for true positives, there is a significant difference between the number of patterns: $F(3, 56) = 6.222$, $p = 0.001$. Specifically, the number of true positives detected with one pattern is different from three patterns at significance level $p = 0.008$, and different from four patterns at significance level $p = 0.002$ (no other pairs of numbers of patterns were statistically different). For false positives and accuracy, there was no significant difference between number of patterns ($p = 0.056$ and $p = 0.764$, respectively).

Table 5. Results for different number of patterns

	True Positives				False Positives				Accuracy			
No. of patt.	A	B	C	Ave.	A	B	C	Ave.	A	B	C	Ave.
1	0.8	0.8	0.6	0.7	0.3	0.1	0.3	0.2	0.6	0.8	0.5	0.6
2	0.8	0.4	0.4	0.5	0.2	0.1	0.3	0.2	0.7	0.6	0.4	0.6
3	0.2	0.2	0.2	0.2	0.2	0.0	0.1	0.1	0.4	0.6	0.5	0.5
4	0.0	0.2	0.2	0.1	0.1	0.0	0.0	0.0	0.4	0.6	0.6	0.5

These values support the conclusion that it is possible to classify unknown texts as instances of a particular view of a previously known domain by using patterns extracted manually from eight examples. They also suggest focusing on a small number of matching semantic patterns as we work towards improving the method's accuracy. We did not evaluate the elapsed time in these runs, though most runs took around one second to run, which is our target speed.

Discussion. As we increase the number of required patterns for detection, it becomes harder to get TP. Analogously, FP also reduce, which is usually good. However, depending on the domain, the impact of reducing FP may be too costly on TP or vice-versa.

Receiver Operating Characteristic (ROC) analysis is meant to provide a visual evaluation of the tradeoffs between TP and FP [4]. The ROC space is represented with TP in the Y axis and FP in the X axis [ibid.]. The ROC analysis of our method is depicted in Figure 44. The growth in TP is accompanied by increasing values for FP.

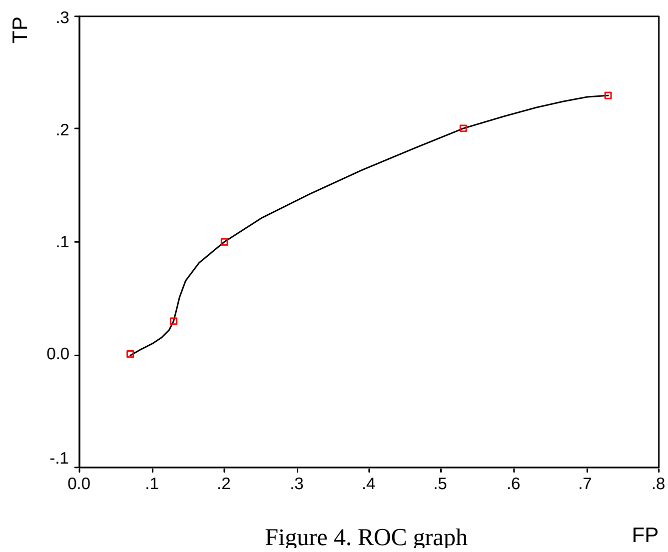

Figure 4. ROC graph

For a more detailed evaluation of the tradeoffs between TP and FP, Table 6 shows the sensitivity of the average (ave.) of absolute numbers of documents that changed classifications. When the number of patterns varies from one to two, there is a reduction of .3 in FP, but its cost is one less TP. This trade-off suggests the minimum of one pattern for the classification. In a domain such as terrorist detection, it is considered preferable to suspect of one more innocent person than to let one guilty terrorist go undetected [1].

Table 6. Sensitivity of TP vs. FP

# patterns	TP average	Difference	FP average	Difference
1	3.7		2.3	
2	2.7	-1.0	2.0	0.3
3	1.0	-1.7	1.0	1.0
4	0.7	-0.3	0.3	0.7

Another result in our study is how accuracy behaves with changes in the number of patterns required for classification. This information is important because it gives a measure of the power of detection of the patterns. Figure 5 shows this relation in the data we studied. Additional studies are necessary for verifying these results because we did not observe consistent behavior in the relation between accuracy and the number of patterns.

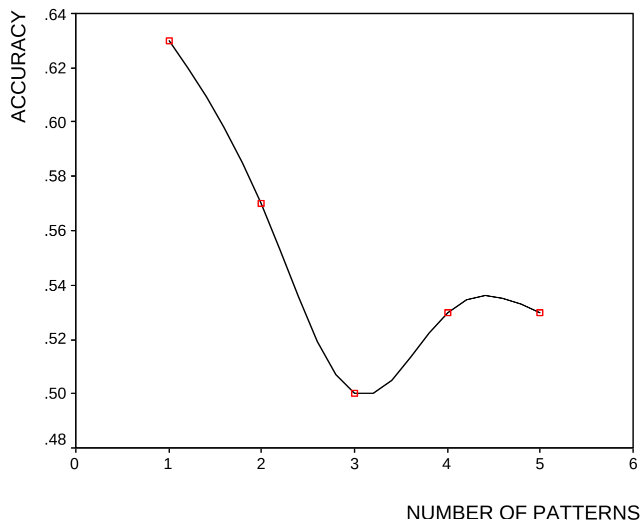

Figure 5. Number of patterns vs. accuracy

5 Conclusions and Future Work

Our goal was to conceive a method that would complement and improve the accuracy of detection of inconspicuous contents obtained by data-intensive methods. Our proposed method relies on a highly simplified pattern representation extracted in a manual engineering step that uses a very small set of training instances. Additionally, we attempted to process and classify ten input texts under one second to support collaboration with real-time data-intensive applications.

The main contribution of this work refers to the development of a classification approach that does not require a training corpus and uses a simplified representation of semantic meaning while potentially bringing data-intensive methods such as [1] from 90% accuracy to 96% accuracy.

The preliminary evaluation of the ability of the manually extracted patterns to classify text shows promise. We found that it is possible to classify unknown texts as instances of a particular view of a previously known domain, with the results suggestive of focusing on a small number of matching semantic patterns while working towards improving accuracy.

The processing time varied between one and two seconds. The entire design of the method has focused on speed but we have not empirically validated the execution time yet.

The main limitations of this work relate to the use of humans in the process. Humans can be expensive and are not usually consistent. Consequently, human judgment may interfere in the quality of the results. Nevertheless, we believe this is a cost worth paying if the final results improve the overall accuracy and may help detect one more web page retaining inconspicuous contents.

The representation of patterns may be used in different textual methods such as textual case-based reasoning. Any classification task where a large training corpus is not available may also benefit from this approach. The real-time detection offered by the aggregated methods that combines data-intensive plus our patterns can be used to detect a variety of inconspicuous contents.

Acknowledgements

We would like to thank Dr. Mark Last for his suggestions on this work. Dr. Rosina Weber, Amit Deshpande, and Jason M. Proctor are supported in part by the National Institute for Systems Test and Productivity at USF under the USA Space and Naval Warfare Systems Command grant no. N00039-02-C-3244, for 2130 032 L0, 2002.

References

[1] Elovici, Y., Kandel, A., Last, M., Shapira, B., Zaafrany, O.: Using Data Mining Techniques for Detecting Terror-Related Activities on the Web. Journal of Information Warfare 3, 1 (2004) 17 – 29
[2] Cohen, W. M., Carvalho, V. R., Mitchell, T. M.: Learning to Classify Email into Speech Acts. In: Lin, D., Wu, D. (ed.): Proc. of the 2004 Conference on Empirical Methods in Natural Language Processing (2004) 309-316
[3] Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. Proc. of the Third IEEE International Conference on Data Mining. IEEE Computer Society (2003)
[4] Provost, F., Fawcett, T.: Robust Classification for Imprecise Environments. Machine Learning 42, 3 (2001) 203-231
[5] Haller, S.: An Introduction to Interactive Discourse Processing from the Perspective of Plan Recognition and Text Planning. Artificial Intelligence Review 13 (1999) 259–311
[6] Marcu, D.: The Theory and Practice of Discourse Parsing and Summarization. MIT Press, Cambridge, Massachussetts (2000)
[7] Branting, L. K. and Lester, J. C.: Justification Structures for Document Reuse. In: Smith, I., Faltings, B. (eds.): Advances in Case-Based Reasoning. LNCS, Vol. 1168 Springer, Berlin (1996) 76-90
[8] Weber, R.: Intelligent jurisprudence research. In: Proc. of the Seventh International Conference on Artificial Intelligence and Law. ACM, New York (1999) 164-172
[9] Ashley, K. D.: Modeling Legal Argument: Reasoning with Cases and Hypotheticals. A Bradford book. The MIT Press, Cambridge, Massachussetts (1990)
[10] Brüninghaus, S., Ashley, K. D.: Bootstrapping Case Base Development with Annotated Case Summaries. In: Althoff, K-D, Bergmann, R., Branting, LK. (eds.): Case-Based Reasoning Research and Applications. LNAI, Vol. 1650. Springer, Berlin (1999) 59-73
[11] Brüninghaus, S., Ashley, K. D.: Reasoning with Textual Cases. In: Munoz, H. and Ricci, F. (eds.): Case-Based Reasoning Research and Applications. LNAI Springer, Berlin (2005)
[12] Chandrasekaran, B., Josephson, J.R., Benjamins, V. R.: What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems 14, 1 (1999) 20-26