New York University - Stern School of Business
Working paper CeDER-11-01

Explaining Documents' Classifications

David Martens
Faculty of Applied Economics, University of Antwerp, Belgium David.Martens@ua.ac.be

Foster Provost
Department of Information, Operations and Management Sciences, Leonard N. Stern School of Business, New York University, NY fprovost@stern.nyu.edu

This is a design-science paper about methods for explaining data-driven classifications of text documents. Document classification has widespread applications, such as with web pages for advertising, emails for legal discovery, blog entries for sentiment analysis, and many more. Document data are characterized by very high dimensionality, often with tens of thousands to millions of variables (words). Many applications require human understanding of the reasons for classification decisions: by managers, client-facing employees, and the technical team. Unfortunately, due to the high dimensionality, understanding the decisions made by the document classifiers is very difficult. Previous approaches to gain insight into black-box models do not deal well with high-dimensional data. Our main theoretical contribution is to define a new sort of explanation, tailored to the business needs of document classification and able to cope with the associated technical constraints. Specifically, an explanation is defined as a set of words (terms, more generally) such that removing all words within this set from the document changes the predicted class from the class of interest. We present an algorithm to find such explanations, as well as a framework to assess such an algorithm's performance. We demonstrate the value of the new approach with a case study from a real-world document classification task: classifying web pages as containing adult content, with the goal of allowing advertisers to choose not to have their ads appear there. We present a further empirical demonstration on news-story topic classification using the 20 Newsgroups benchmark dataset. The results show the explanations to be concise and document-specific, and to provide insight into the exact reasons for the classification decisions, into the workings of the classification models, and into the business application itself. We also illustrate how explaining documents' classifications can help to improve data quality and model performance.

Key words: Document Classification, Instance Level Explanation, Text mining, Comprehensibility

---

1. Introduction

Document classification aims to classify textual documents automatically, based on the words, phrases, and word combinations therein (hereafter, "words"). Business applications of document classification have seen increasing interest, especially with the introduction of low-cost micro-outsourcing systems for annotating training corpora. Prevalent applications include sentiment analysis (Pang and Lee 2008), spam identification (Attenberg et al. 2009), and web page classification (Qi and Davison 2009), just to cite a few. Classification models are built from labeled data sets that

encode the frequencies of the words in the documents. Importantly for this paper, and different from many data mining applications, the document classification data representation has very high dimensionality, with the number of words and phrases typically ranging from tens of thousands to millions.

The main contribution of this design-science paper is to extend substantially our understanding and capability with respect to an important aspect of the business application of document classification, an aspect that has received little attention in the research literature. Specifically, organizations often need explanations for the exact reasons why classification models make particular decisions. The need comes from various perspectives, including those of managers, customer-facing employees, and the technical team. Customer-facing employees need to deal with customer queries regarding the decisions that are made; it often is insufficient to answer that the magic box said so. Managers need to "sign off" on models being placed into production, and want to understand how the model makes its decisions, rather than just to trust the technical team or data science team.[1] Managers also need to understand specific decisions when they are called into question by customers or other managers.

Finally, the technical/data science team itself needs to understand the reasons for decisions in order to be able to debug and improve the models. Wholistic views of a model and aggregate statistics across a "test set" may not give sufficient guidance as to how the model can be improved. The instance-level explanation methods introduced in this paper can have a substantial impact in improving the process of building document classification models. Despite the stated goals of early research on data mining and knowledge discovery (Fayyad et al. 1996), very little work has addressed support for the process of building acceptable models, especially in business situations where various parties must be satisfied with the results.

As a concrete illustration, consider an application currently receiving substantial interest in on-line advertising: keeping ads off of objectionable web content (eMarketer April 27, 2010). Having invested substantially in their brands, firms cite the potential to appear adjacent to nasty content as the primary reason they do not spend more on on-line advertising. To help reduce the risk, document classifiers are applied to web pages along various dimensions of objectionability, including adult content, hate speech, violence, drugs, bomb-making, and many others. However, because the on-line advertising ecosystem supports the economic interests of both advertisers *and* content publishers, black-box models are insufficient. Managers cannot put models into production that

[1] Different applications have different degrees of need for explanations to customers, with denying credit or blocking advertisements being at one extreme. However, even in applications for which black-box systems are deployed routinely, such as fraud detection (Fawcett and Provost 1997), *managers* still need to have confidence in the operation of the system and may need to understand the reasons for particular classifications when errors are made.

might block advertising from substantial numbers of non-objectionable pages, without understanding the risks and incorporating them into the product offering. Customer-facing employees need to explain why particular pages were deemed objectionable by the models. And the technical team needs to understand the exact reasons for the classifications made, so that they can address errors and continuously improve the models.

Popular techniques to build document classification models include naive Bayes, linear and non-linear support vector machines (SVMs), classification-tree based methods (often used in ensembles, such as with boosting (Schapire and Singer 2000)), and many others (Hotho et al. 2005). Because of the massive dimensionality, even for linear and tree-based models, it is very difficult to understand exactly how a given model classifies documents. It is essentially impossible for a non-linear SVM or an ensemble of trees.

Understanding particular classifications also provides important secondary benefits. Not only do we get insight into the classification model, the explanations can provide a novel lens into the complexity of the business domain. For example, in Explanation 1 (shown below; described fully in Section 2.3), the word 'welcome' as an indication of adult content initially seems strange. Upon reflection/investigation we understand that in some cases an adult website's first page contains a phrase similar to *'Welcome to ... By continuing you confirm you are an adult and agree with our policy'*. The explanation brings this complexity to light. We also learn about the various different sub-topics that comprise the class of interest. For example, we find (curious) foreign language adult pages—whose infrequent presence can be lost in the massive dimensionality.

On the theory side, this paper provides a new problem definition and describes the specific characteristics of the problem that differentiate it from those addressed in prior research. Specifically, we focus on explaining why a document is classified as a specific class of interest (e.g., "objectionable content" or "hate speech"). We also discuss what are the important dimensions for evaluating such an explanation-producing system. Based on this framework, we then introduce the first (to our knowledge) technique that directly addresses the explanation of the decisions made by document classifiers. We demonstrate the method empirically, conducting a case study on data from a real application to the business problem of safe advertising discussed above. We augment the case study with an empirical follow-up study on benchmark data sets (news classification). These studies demonstrate that the methods can be fast and effective. The studies also flush out additional important issues in explaining document classifications, such as the need for hyper-explanations (defined below).

Explanation 1: An example explanation why a web page is classified as having adult content.

If words (welcome fiction erotic enter bdsm adult) are removed then class changes from adult to non-adult.

2. Explaining Documents' Classifications

Prior research has examined two different sorts of "explanation" procedures for understanding predictive models: global explanation and instance-level explanation. Global explanations provide insight into the complete model, and its performance over the entire space of possible instances. Instance-level explanations provide explanations for the model's classification of an individual instance—which is our focus here. We now will describe why existing methods are not ideal (or not suitable) for explaining document classification, and then present a new approach that addresses the drawbacks. First, let's discuss the relevant aspects of document classification.

2.1. Key Aspects of Document Classification

As digital text document repositories proliferate and grow, the automated analysis of text documents becomes both an opportunity and a requirement—with our safe advertising example illustrating both. Text mining has been defined as the *"application of algorithms and methods from the fields machine learning and statistics to texts with the goal of finding useful patterns"* (Hotho et al. 2005). We specifically focus on textual document classification, where the value of a discrete target variable is predicted based on the values of a number of independent variables representing the words.[2]

There are several ways in which document classification differs from traditional data mining for common applications such as credit scoring, medical diagnosis, fraud detection, churn prediction and response modeling. Firstly, the data instances have less structure. Specifically, an instance is simply a sequence of words and for most document classification applications the sequential structure is ignored, resulting in simply a bag (multiset) of words. In contrast, traditionally classifier induction has been applied to structured data sets, where each instance for classification is represented as a feature vector: a row from a database table with the values for a fixed number of variables. Technically, one can engineer a feature representation from the sequence or bag of words, but this leads us to our second main difference. In a feature-vector representation of a document data set, the number of variables is the number of words (phrases, n-grams, etc.), which is orders of magnitude larger than in the "standard" classification problems presented above. Thirdly, the

[2] Technically, text document classification applications generally use "terms" that include not only individual words, but phrases, n-grams, etc. For this paper, we call all these "words." Cases where the terms actually are not comprehensible to a human present a limitation of our approach.

values of the variables in a text mining data set denote the presence, frequency of occurrence, or some positively weighted frequency of occurrence of the corresponding word (see below).

These three aspects of document classification all are critical for the explanation of classifier decisions. The first two combine to render existing explanation approaches relatively useless (as we discuss in detail next). The third, however, presents the basis for the design of the solution we propose. Specifically, with all such document classification representations, removing words always corresponds to reducing the value of the corresponding variable or setting it to zero.

A few technical details of document classification are important to understand the techniques we introduce. As preprocessing, all non-textual symbols, such as punctuations, spaces or tabs, are removed from each document. The set of the different words present in any of the documents, constitutes the dictionary. For a set of n documents and a vocabulary of m words, a data set of $n \times m$ is created with the value on row i and column j denoting the frequency of word j in document i. As such, each document is described by a numerical row vector. As most of the words available in the vocabulary will not be present in any given document, most values will be zero, and a sparse representation is used. Often a weighting scheme is applied to the frequencies, where the weights reflect the importance of the word for the specific application (Hotho et al. 2005). A commonly used data-driven weighting scheme is *tfidf*: $x_{ij} = \mathrm{tf}_{ij} \times \mathrm{idf}_j$ where the weight of a word is the "inverse document frequency," which describes how uncommon the word is: $idf(w_j) = log(n/n_j)$ with n_j the number of documents that contain word w_j.

Classification models are built using a training set of labeled documents, where "labeled" means that for the training set we know the value of the "target" variable (the dependent variable being predicted/estimated). The resultant classification model, or classifier, maps any document to one of the predefined classes, and more specifically generally maps it to a score representing the likelihood of belonging to the class, and this score is compared to a threshold for classification. Based on an independent test set, the performance of the model can be assessed by comparing the true label with the predicted label.

2.2. Global explanations

The most common approach to understanding a predictive model is to examine the coefficients of a linear model. Unfortunately such an approach is impracticable for a model with 10^4 to 10^6 variables. For such applications, the most common approach for a linear model is to list the variables (words in our case) with the highest weights. To understand more complex models such as neural networks (Bishop 1996) and non-linear support-vector machines (SVMs) (Vapnik 1995), the principle approach is rule extraction: rules or trees are extracted that mimic the black box as closely as possible (Craven and Shavlik 1996, Martens et al. 2009). The motivation for using

rule extraction is to combine the desirable predictive behavior of non-linear techniques with the comprehensibility of decision trees and rules. Previous benchmarking studies have revealed that when it comes to predictive accuracy, non-linear methods often outperform traditional statistical methods such as multiple regression, logistic regression, naive Bayesian and linear discriminant analysis (see e.g. Baesens et al. (2003), Lessmann et al. (2008)). For some applications however, e.g., medical diagnosis and credit scoring, a clear explanation of how the decision is reached by models obtained by these techniques is a crucial business requirement and sometimes a regulatory requirement.

The baseline rule extraction approach is to replace the given class labels of all data instances with those provided (predicted) by the black box model. By applying a rule or tree induction technique on this new data set, the resulting model will be a comprehensible tree or rule set that explains the functioning of the black box model. Generally the complexity of the tree or rule set increases with its fidelity—the proportion of instances for which the extracted rules make the same prediction as the black box model.

These rule extraction approaches are not suitable for our present problem for several reasons. Not all classifications are explained by these rule extraction approaches (as we will demonstrate for the most common approach). Additionally, for some instances that seem to be explained by the rules, more refined explanations exist. In addition, often one is only interested in the explanation of the classification of a single data instance—for example, because it has been brought to a manager's attention because it has been misclassified or simply because additional information is required for this case.

In addition, global explanations do not provide much insight for document classification anyway, because of the massive dimensionality. For a classification tree to remain readable it can not include thousands of variables (or nodes). Similarly, listing all these thousands of words with their corresponding weights for a linear model will not provide much insight into individual decisions. Clearly an explanation approach focusing on individual classifications would be preferred. Considering our running example of web page classification for safe advertising, what we want to know is *'Why did the model classify this web page as containing objectionable content?'*

2.3. Instance-level explanations

Over the past few years, instance explanation methods have been introduced that explain the predictions for individual instances[3] (Robnik-Šikonja and Kononenko 2008, Štrumbelj et al. 2009, Štrumbelj and Kononenko 2010, Baehrens et al. 2010). Generally, these methods provide a real-valued score to each of the variables that indicates to what extent it contributes to the data

[3] The technical details of the prior approaches are described in the Appendix.

instance's classification. This definition of an explanation as a vector with a real-valued contribution for each of the variables makes sense for many classification problems, which often have relatively few variables (e.g. the median number of variables for the popular UCI benchmark datasets is 18.5 (Hettich and Bay 1996)). For document classification, however, due to the high-dimensionality of the data, this sort of explanation is not ideal—and possibly not useful at all. Considering our safe-advertising data set, an explanation for a web page's classification as a vector with thousands of non-zero values can hardly be considered comprehensible. Although the words with the highest contributions will have the biggest impact on the classification, we still don't know which (combination of) words actually led to any given classification.

Aside from the unsuitable format of these previous explanations, previous instance-based explanation approaches are unable to handle high dimensional data computationally. The sample-based approximation method of Štrumbelj and Kononenko (2010) is reported to be able to handle up to about 200 variables—even there requiring hours of computation time. The authors acknowledge that for such data sets other approaches should be introduced:

> *Arguably, providing a comprehensible explanation involving a hundred or more features is a problem in its own right and even inherently transparent models become less comprehensible with such a large number of features* (Štrumbelj and Kononenko 2010).

Because of this inability to deal with the high-dimensionality of text mining data sets, as well as the explanation format as a real-valued vector, these methods are not applicable for explaining documents' classifications.

In focusing on document classification, we take advantage of three main observations to define a slightly different problem from that addressed by prior work, that will address the motivating business needs and that we will be able to solve efficiently. The first observation is that in many document classification problems there really are two quite different explanation problems. We often are interested specifically in one of them: why documents were classified as a particular focal class (a "class of interest"). Considering our web page classification setting, we will focus primarily on explaining why a page has received (rightly or wrongly) a "positive" classification of containing objectionable content. The asymmetry is due to the negative class being a default class: if there is no evidence of the class of interest (or of any of the classes of interest), then the document is classified as the default class. In this paper we will not treat in detail the other explanation problem. The question of why a particular page has *not* received a positive classification can be important as well, but reflection tells us that it is indeed a very different problem. Often the answer is "the page did not exhibit any of the countless possible combinations of evidence that would have led the model to deem it objectionable." The problem here generally is "how do I *fix* the model given that I believe it has made an error on this document." This is a fundamentally different

problem and thereby should require a very different solution—for example, an interactive solution where users try to explain to the system why the page should be a positive, for example using dual supervision (Sindhwani and Melville 2008), or a relevance feedback/active learning systems where chosen cases are labeled and then the system is retrained. These are important problems, but are beyond the scope of this paper.

The second important observation is that in contrast to the individual variables in many predictive modeling tasks, individual words can be quite comprehensible. Thus for us an explanation will be a set of words present in the document such that removing all occurrences of these words results in a different classification (defined precisely below). The innate comprehensibility of the words often will immediately give deep intuitive understanding of the explanation. As we will see, when it does not it can indicate problems with the model. Under this definition, we see that we may be interested in the minimal explanation or the set of minimal explanations for a document. We will return to this below.

The third observation is that in document classification, removing all occurrences of a word always sets the corresponding variable's value to zero. This will allow us to formulate an optimization problem for which we can find solutions fast.

2.4. Explaining the Classification of Documents

As discussed above, the question we address is *'Why is this document classified as the non-default (here adult content) class?'* To answer this question we provide an explanation as a set of words present in the document such that removing these words causes a change in the class. Only when all the words in the explanation are removed does the class change, and as such the set is minimal.

To define an explanation formally (see Definition 1) we need to recall that a document $D \in \mathcal{D}$ is a bag (multiset) of words. Let W_D be the corresponding set of words. We presume that classifications are based on a classifier C_M, which is a function from documents to classes. Later, our heuristic algorithm will presume that C_M incorporates at least one scoring function f_{C_M}; classifications will be based on scores exceeding thresholds (in the binary case), or choosing the class with the highest score (in the multiclass case). The majority of classification algorithms operate in this way, including all that we discuss in this paper.

Definition 1. Given a document D consisting of m_D unique words W_D from the vocabulary of m words: $W_D = \{w_i, i = 1, 2, \ldots, m_D\}$, which is classified by classifier $C_M : \mathcal{D} \rightarrow \{1, 2, \ldots, k\}$ as class c. We define an *explanation for document D's classification* as a set E of words such that removing all words in E from the document leads C_M to produce a different classification. Further, an explanation E is minimal in the sense that removing any subset of E does not yield a change in class. Specifically:

E is an explanation for $C_M(D) \iff$
1. $E \subseteq W_D$ (the words are in the document),
2. $C_M(D \backslash E) \neq c$ (the class changes), and
3. $\nexists E' \subset E | C_M(D \backslash E') \neq c$ (E is minimal).
$D \backslash E$ denotes the result of removing the words in E from document D.

Definition 1 is specifically tailored to document classification. It provides intuitive explanations in terms of words present in the document, and we will be able to produce such explanations even in the massively dimensional input spaces typical of document classification. More specifically, Definition 1 differs from those of prior approaches in that the explanation is a set of words rather than a vector. Define the size of the explanation as the cardinality of E. Our empirical analysis will reveal that explanations typically are quite small (often about a dozen words) and as such the technique is able to effectively transform the high-dimensional input space to a low-dimensional explanation. As stated before, this is of crucial importance in order to provide insightful explanations that address the business problems at hand, i.e. managers' needs to understand classifiers' behavior, explaining the decisions made to the manager or customer, obtaining insights into the specific domain, or improving the document classification model's performance.

The goal of the present approach seems to align with that of inverse classification (Mannino and Koushik 2000). However, the explanation format, the specific optimization problem, and the search algorithms are quite different. Firstly, for document classification, we only need to consider reducing the values for the corresponding variables. Increasing the value of variables does not make sense in this setting. For example, in the case of classifying web pages as having adult content or not, simply adding words as 'xxx' would likely increase the probability of being classified as adult. This is valid for all documents and does not really explain the document's classification. Secondly, we don't need to decide on step sizes for changes in the values, as removing the occurrences of a word corresponds to setting the value to zero. In the optimization routine of inverse classification, the search is exactly finding the minimal distance for each dimension. The optimization is completely different for explanations of documents' classification, as we will discuss next. Thirdly, applying inverse classification approaches to document classification generally is not feasible, due to the huge dimensionality of these data sets. Our approach takes advantage of the sparseness of document representations, and only needs to consider those words actually present in the document. Finally, we provide a general framework to obtain explanations independent of the classification technique used.

The desire to be model-independent is important and worth discussing further. For document classification, non-linear, black-box models are often used, such as non-linear SVMs (Joachims

1998) or boosted trees (Schapire and Singer 2000). These models are often incomprehensible. Explaining the decisions made by such techniques to a client, manager, or subject-matter expert is of great value and a natural application of our framework. When a linear model is being used, one could argue simply to list the top k words that appear in the document with the highest positive weights as an explanation for the class (assuming we are explaining class 1 versus class 0). The choice of k can be set to 10 for example. A more suitable choice for k would be the minimal number of top words such that removing these k words leads to a class change. This is exactly what our approach would provide with a linear model. Finally, although they are often cited as producing comprehensible models, classification trees for document classification do not provide the sort of explanations we need (as in Definition 1): they do not explain what words actually are responsible for the classification. All words from the root to the specific leaf for this document may be important for the classification, but some of these words are likely not present in the document (the path branched on the absence of the word) and we do not know which (minimal) set of words actually is responsible for the given classification.

3. Finding Document Classification Explanations

The discussion above allows us to understand the problem more precisely from an optimization perspective. Unlike the settings in prior work, here we are looking for the shortest paths in the space defined by word *presence*, based on the effect on the surface defined by the document classification model—which is in a space defined by more sophisticated word-based features (e.g., frequency or tfidf, as described above). Conceptually, given a document vocabulary with m words, consider a *mask vector* μ to be a binary vector of length m, with each element of the vector corresponding to one word in the vocabulary. An explanation E can be represented by a mask vector μ_E with $\mu_E(i) = 1 \iff w_i \in E$ (otherwise, $\mu_E(i) = 0$). Recall that the size of the explanation is the cardinality of E, which becomes the L1-norm of μ_E. Then $D \backslash E$ is the Hadamard product of the feature vector of document D (which may comprise frequencies or tfidf values) with the one's complement of μ_E.[4] Thus, finding a minimal explanation corresponds to finding a mask vector μ_E such that $C_M(D \backslash E) \neq C_M(D)$ but if any bit of μ_E is set to zero to form E', $C_M(D \backslash E') = C_M(D)$.

To our knowledge, this sort of explanation for document classification has not previously been formalized or examined carefully, so before presenting algorithms for producing document explanations, we should discuss the possible objectives precisely.

[4] In the case of a binary D, this simply becomes a bitwise NAND of D and μ_E.

3.1. Objectives and Performance Metrics

Although Definition 1 is quite concise, the objectives for an algorithm searching for such explanations can vary greatly. A user may want to: (1) Find a minimum-size explanation: an explanation such that no other explanation of smaller size exists. (2) Find all minimal explanations. (3) Find all explanations of size smaller than a given k. (4) Find l explanations, as quickly as possible ($l = 1$ may be a common objective). (5) Find as many explanations as possible within a fixed time period. Combinations of such objectives may also be of interest. To allow the evaluation of different explanation procedures for these objectives, we must define a set of performance metrics[5]:

Search effectiveness:

1. PE: Percentage of test instances explained (%)
2. ANE: Average number of explanations given (number)

Explanation complexity:

3 AWS: Average number of words in the smallest explanation (number)

Computational complexity:

4 ADF: Average duration to find first explanation (seconds)

5 ADA: Average duration to find all explanations (seconds)

These performance metrics describe the behavior of a document explanation algorithm. In a separate analysis, one can also employ a domain expert to verify the explanations. An interesting question that is beyond the scope of this paper is: if the explanations are counterintuitive, does that reflect on the explanation-finding method? Or only on the underlying classification model that is being explained? We will show that some explanations reveal the overfitting of the training data by the modeling procedure—which often is not revealed by traditional machine learning evaluations that examine summary statistics (error rate, area under the ROC curve, etc.).

3.2. Complete Enumeration of Explanations of Increasing Size

A naive approach to producing explanations completely enumerates all word combinations, starting with one word, and increasing the number of words until an explanation is found. This approach starts by checking whether removing one word w from the document would cause a change in the class label. If so, we add the explaining rule '**if** word w is removed **then** the class changes'. We check this for all of the words that are present in the document. For a document with m_D words, this requires m_D evaluations of the classifier. If the class does not change based on one word only, the case of several words being removed simultaneously will be considered. First, the algorithm

[5] Note that explanation accuracy is not a major concern: as an explanation by definition should change the predicted class, it is straightforward to ensure that explanations produced always are correct. What is important with regards to the usefulness of an explanation (or set of explanations) is how complex the explanation is, and how long it took for the algorithm to find the explanation.

considers all word combinations of size 2, then 3 and so on. For combinations of 2 words, the algorithm makes $m_D \times (m_D - 1)$ evaluations, for all combination of 3 words $m_D \times (m_D - 1) \times (m_D - 2)$ evaluations, and more generally for combinations of k words we need $m_D!/(m_D - k)! = O(m_D^k)$ evaluations. This scales exponentially with the number of words in the document, and becomes infeasible for real-world problems.

3.3. Explaining Documents' Classifications: A Hill-Climbing Approach

As the number of potential explanations scales exponentially with the number of features, the naive approach cannot be applied to realistic problems. We now introduce a straightforward, heuristic approach, formally described in Algorithm 1. It is designed specifically to find a solution in reasonable time, even though solution might not be the optimal, in the sense that smaller explanations could exist. (We will see that it indeed is optimal in a certain, important setting.) The approach is based on two notions:

1. **Hill-climbing search:** We assume that the underlying classification model will always be able to provide a probability estimate or score[6] in addition to a categorical class assignment. We will denote this score function for classifier C_M by $f_{C_M}(\cdot)$. The algorithm starts by listing all potential explanations of one word, and calculating the class and score change for each. The algorithm proceeds as a straightforward hill-climbing search. Specifically, at each step in the search, given the current set of word combinations denoting partial explanations, the algorithm next will expand the partial explanation for which the output score changes the most in the direction of class change. Expanding the partial explanation entails creating a set of new, candidate explanations, comprising all combinations with one additional word from the document (that is not yet included in the partial explanation).

2. **Pruning:** For each explanation with l words that is found, we do not need to check combinations of size $l + 1$ with these same words, hence we can prune these branches of the search tree. For example if the words 'hate' and 'furious' provide an explanation, we are not interested in explanations of three words that include these two words, such as 'hate', 'furious' and 'never'.

For the case of a linear classifier with a binary feature representation, we might explain the classification by looking at the words with the highest weights that appear in the document. However, we would still want to know which words exactly are responsible for the classification. The proposed SEDC produces optimal (minimum-size) explanations for linear models, which we discuss further next. Assuming again a class 1 versus class 0 prediction for document j, SEDC

[6] No explicit mapping to [0, 1] is necessary; a score that ranks by likelihood of class membership is sufficient. The scores for different classes must be comparable in the multiclass case, so in practice scores often are scaled to [0,1]. For example, support-vector machines' output scores are often scaled to (0,1) by passing them through a simple logistic regression (Platt 1999).

Algorithm 1 SEDC: Search for Explanations for Document Classification (via Hill Climbing with Pruning)

Inputs:
$W_D = \{w_i, i = 1, 2, \ldots, m_D\}$ % Document D to classify, with m_D words
$C_M : \mathcal{D} \rightarrow \{1, 2, \ldots, k\}$ % Trained classifier C_M with scoring function f_{C_M}
$max\_iteration$ % Maximum number of iterations
Output:
Explanatory list of rule R

$c = C_M(D)$ % The class predicted by the trained classifier
$p = f_{C_M}(D)$ % Corresponding probability or score
$R = \{\}$ % The explanatory list that is gradually constructed
$combinations\_to\_expand\_on =$ set of all words
$P\_combinations\_to\_expand\_on$
for all words w in $combinations\_to\_expand\_on$ **do**
  $V_w = 0$; % As if the word did not appear in the document
  $c_{new} = C_M(D \cup V_W)$ % The class predicted by the trained classifier if the word w did not appear in the document
  $p_{new} = f_{C_M}(D \cup V_W)$ % The probability or score predicted by the trained classifier if the word w did not appear in the document
  $P\_combinations\_to\_expand\_on = P\_combinations\_to\_expand\_on \cup p_{new}$
  if $c_{new} \neq c$ **then**
    $R = R \cup$ '*if* word w is removed *then* class changes'
    $combinations\_to\_expand\_on =$ remove word w from $combinations\_to\_expand\_on$
  end if
end for
for $iteration = 1$ to $max\_iteration$ **do**
  $combo$ = word combination in $combinations\_to\_expand\_on$ for which $p - p\_combinations\_to\_expand\_on$ is maximal
  $combo\_set$ = create all expansions of $combo$ with one word
  $combo\_set2$ = remove explanations from $combo\_set$
  $p\_combo\_set2 = \{\}$
  for all combos Co in $combo\_set2$ **do**
    for all words w_j in Co **do**
      $V_{w_j} = 0$; % As if the word did not appear in the document
    end for
    $c_{new} = C_M(D \cup V_W)$ % The class predicted by the trained classifier if the words W did not appear in the document
    $p_{new} = f_{C_M}(D \cup V_W)$ % The probability or score predicted by the trained classifier if the words W did not appear in the document
    $p\_combo\_set2 = p\_combo\_set2 \cup p_{new}$
    if $c_{new} \neq c$ **then**
      $R = R \cup$ '*if* words W are removed *then* class changes'
      $combo\_set3$ = remove explanation in R from $combo\_set2$
    end if
  end for
  $combinations\_to\_expand\_on = combinations\_to\_expand\_on \cup combo\_set3$
  $P\_combinations\_to\_expand\_on = P\_combinations\_to\_expand\_on \cup p\_combo\_set2$
end for

ranks all words appearing in the document according to the product $w_j x_{ij}$. An explanation of smallest size is the one with the top-ranked words, as chosen by SEDC's hill-climbing search.

LEMMA 1. *For document representations based on linear binary-classification models* $f_{C_M}(D) = \beta_0 + \sum \beta_j x_{ij}$ *with binary (presence/absence) features, the smallest explanation found by SEDC will be a minimum-size explanation. More specifically, for* E_1, E_2 *explanations, if* E_1 *is the smallest explanation found by SEDC,* $|E_1| = k \Rightarrow \nexists E_2 : |E_2| < k$. *Furthermore, the first explanation found by SEDC will be of size* k.

Proof (by contradiction): If no explanation exists, then the theorem holds vacuously. Assume there exists at least one explanation. In the linear model, let the (additive) contribution w_{ij} to the output score for word j of document i be the linear model weight β_j corresponding to binary word-presence feature x^b_{ij} for those words that are present in document i (and zero otherwise).

Assume w.l.o.g. that the classification threshold is placed at $f_{C_M}(D) = 0$. SEDC will compose the first candidate explanation E^* by first selecting the largest w_{ij} such that the word is present in the document, $x^b_{ij} = 1$, and adding word j to the explanation. SEDC will then add to E^* the word with the next-largest such w_{ij}, and so on until $f_{C_M}(E^*) \leq 0$. Thus, the first explanation E_1 by construction will consist of the k highest-weight words that are present in the document.

Now assume that there exists another explanation E_2 such that $|E_2| < k$; being an explanation, $f_{C_M}(E_2) \leq 0$. Recall that explanations are minimal, so $\nexists S \subsetneq E_1 : f_{C_M}(S) \leq 0$. Thus E_2 must have at least one element $e \notin E_1$. Let $\sum_E$ denote the sum of the weights corresponding to the words in an explanation E. For a linear model based on the (binary) presence/absence of words, $f_{C_M}(X \backslash Y) = f_{C_M}(X) - \sum_Y$. As noted above, E_1 comprises by construction the k words with the largest w_{ij}, so $\forall w_{ij} \in E_1, \forall w_e \notin E_1 : w_{ij} \geq w_e$. Therefore, $\exists S \subsetneq : E_1, \sum_S > \sum_{E_2}$, which means that $\exists S \subsetneq E_1 : f_{C_M}(D \backslash S) \leq f_{C_M}(D \backslash E_2)$. But $\forall S \subsetneq E_1 : f_{C_M}(D \backslash S) > 0$ and thus $f_{C_M}(D \backslash E_2) > 0$. Therefore, E_2 is not an explanation, a contradiction. □

This optimality applies as well to monotonic transformations over the output of the linear model, as with the common logistic transform used to turn linear output scores into probability estimates. The optimality also applies more generally for linear models based on numeric word-based features, such as frequencies, tfidf scores, etc., as detailed in the following theorem.

THEOREM 1. *For document representations based on linear models* $f_{C_M}(D) = \beta_0 + \sum \beta_j x_{ij}$ *with numeric word-based features, such as frequencies or tfidf scores, that take on positive values when the word is present and zero when the word is absent, the smallest explanation found by SEDC will be a minimum-size explanation. More specifically, for* E_1, E_2 *explanations, if* E_1 *is the smallest explanation found by SEDC,* $|E_1| = k \Rightarrow \nexists E_2 : |E_2| < k$. *Furthermore, the first explanation found by SEDC will be of size* k.

Proof: Decompose each non-negative word feature x_{ij} into the product $x^b_{ij}d_{ij}$ of a binary word presence/absence feature x^b_{ij} and a document-specific non-negative weight d_{ij}. The corresponding term in the linear model $\beta_j x_{ij}$ then becomes $\beta_j d_{ij} x^b_{ij}$. The proof then follows the previous proof directly, except with the additive contribution of each word being $w_{ij} = \beta_j d_{ij}$. □

For non-linear models no such optimal solutions are guaranteed, in the sense that smaller explanations could exist. However, as our empirical work will show, still very good results are obtained, both in search effectiveness, and explanation and computational cost. For multiclass classification problems optimal solutions are also not guaranteed if one decomposes the problem in several binary classification problems (as in a one-versus-rest or one-versus-one approach) since the final classification of data instances now depends on several models with their own weights.

4. Empirical Analysis

We now will demonstrate the value of the approach to explaining document classifications through two, related empirical analyses (Hevner et al. 2004): classifying web pages as containing adult content and news-story topic classification. First we will examine in detail a case study application of the method to a data set drawn from a real application in need of exactly this sort of evaluation. The empirical results show that the method indeed can produce explanations effectively, and that alternative, global explanation techniques do not. Possibly more interestingly, the case study highlights various sorts of practical value that can be obtained from producing model-and-document-specific explanations. We follow-up the case study with a shallower but broader experimental analysis based on a suite of text classification problems (the 20 Newsgroups) widely used in the research literature. The followup analysis highlights how document-specific explanations can help to understand the behavior (and confusion) of a classification model that distinguishes between multiple classes, and more deeply, shows that different sub-categories receive very different explanations. In retrospect, that is not surprising; however, it would be very difficult to ascertain from prior explanation procedures (in particular, global ones). In all, the empirical analysis is intended to demonstrate that explaining document classification with SEDC is capable of (1) providing important insights into the model for the manager and the customer, (2) providing insight into the business domain, and (3) identifying opportunities for model improvement.

4.1. Explaining Web Pages' Classifications for Safe Advertising

The case study is based on data obtained from a firm that focuses on helping advertisers to avoid inappropriate adjacencies between on-line advertisements and web content, similar to our motivational example above. Specifically, the analysis is based on a data set of 25,706 web pages, labeled as either having adult content or not. The web pages are described by tfidf scores over a vocabulary chosen by the firm, including a total of 73,730 unique words. The data set is balanced

by class, with half of the pages containing adult content and half non-adult content. For this data set, the class labels were obtained from a variety of sources used in practice, including Amazon's Mechanical Turk.[7] Given the variety of labeling sources, the quality of the labeling might be questioned (Sheng et al. 2008). Interestingly, the explanations indeed reveal that certain web pages are wrongly classified. No meta-data, links, or information on images is being used for this study; the inclusion of such data could improve the model further, but the focus of this paper is on textual document classification.[8]

For this analysis, we built an SVM document classification model with a linear kernel function using the LIBLINEAR package (Fan et al. 2008), with 90% of the data used as training data, the remaining 10% as test data. Experiments were run on an Intel Core 2 Quad (3 GHz) PC with 8GB RAM. The model is correct on 96.2% of the test instances, with a sensitivity (percentage of non-adult web pages correctly classified) of 97.0%, and a specificity (percentage of adult web pages correctly classified) of 95.6%. The resulting model is a linear function with 73,730 weights (and an intercept term), one for each of the words, clearly calling into question the potential for gaining deep insight into the model's behavior simply by examining it. Some technique is necessary for helping to explain the model.

4.1.1. Global explanations As discussed above, rule extraction is the most researched and applied model explanation methodology. Trying to comprehend the SVM model, a tree can be extracted by applying the C4.5 tree induction technique (Quinlan 1993) on the aforementioned safe advertising data set with class labels changed to SVM predicted labels. Unfortunately, we could not get C4.5 to generate a tree that models the SVM with high-fidelity. The best extracted tree has a fidelity of only 87%. On top of that, the tree is too large to be comprehensbile, having 327 nodes. Pruning the tree further reduces the size, but further decreases fidelity.

As discussed above, an alternative method for comprehending the function of a linear document classifier is to examine the weights on the word features, as these indicate the effect that each word has on the final output score. As with the distinction between Lemma 1 and Theorem 1, we need to keep in mind that in a preprocessing step the data set is encoded in tfidf format. Hence for actual document explanations, the frequency is vital.[9] Figure 1 shows the weight sizes of all the words in the vocabulary; the weights are ranked smallest-to-largest, left-to-right. Clearly many words show a high indication of adult content, while many others show a clear counter-indication

[7] www.mturk.com

[8] This particular data set was not necessarily used in the development of any production model used for safe advertising.

[9] The inverse document frequency is constant across documents, and could be incorporated in the model weights to facilitate global explanation.

of adult content. Looking deeper, Table 1 shows the highest (positive) weight words, as well as the words that give the highest mutual information (with the positive class) and information gain. We additionally list the top words when taking into account the idf weights, viz., based on the weights of the words multiplied with the corresponding idf values. The final column shows the words most frequently occurring in the explanations, which will be elaborated on below. Table 2 shows the ranks of some adult-indicative words provided independently by a domain expert.

From Table 1 we see that most indicative words for adult content ranked highly using the mutual information criterion are very rare, unintuitive words. It may be possible to engineer a better information-based criterion, for example countering this overfitting behavior by requiring a minimal frequency of the top ranked words, but later results will show why such efforts ultimately are destined to fail to provide a comprehensive explanation. The top words provided by the other rankings on the other hand are quite intuitive. As stated before, even initially not-so-obvious words as 'welcome', 'enter' or 'age' make sense once we realize that many positive examples are entrance pages of adult sites, which inform a visitor about the content of the website and require verification of age. Nevertheless, as we will see next, explanation of individual decisions simply requires too many individual words. Consider that we would have to produce a list of over 700 of the highest-weight words just to include 'porn' and over 10,000 to include 'xxx'—two of the short-list of words chosen by the domain expert.

Given the intuitiveness of the top-weighted words, we should consider how well a short list of such words really explains the behavior of the model. Does the explanation of a web page typically consist of (some of) the top-100 or so words? It turns out that the content of web pages varies tremendously, even within individual categories. For "adult content", even though some strongly discriminative words exist, the model classifies most web pages as being adult content for other reasons. This is demonstrated by Figure 2, which plots the percentage of the classifications of the test instances that would be explained by considering the top-k words (horizontal axis) by weight (with and without idf correction), mutual information and information gain. Specifically, if a definition in the sense of Definition 1 can be formed by any subset of the set of top-k words, then the document is explained. So for example, if an explanation would be 'if words (welcome enter) are removed then class changes', that explanation would be counted when $k \geq 2$.

We see from Figure 2 that we would need thousands of these top words before being able to explain a large percentage of the individual documents, as shown by the line with words ranked on the weight. More precisely, more than two thousand top-weight words are needed before even half of the documents are explained. Using the ranking based on mutual information requires even more words. This suggests either (i) that many, many words are necessary for individual explanations, or (ii) the words in the individual explanations vary tremendously. This motivates the use of an

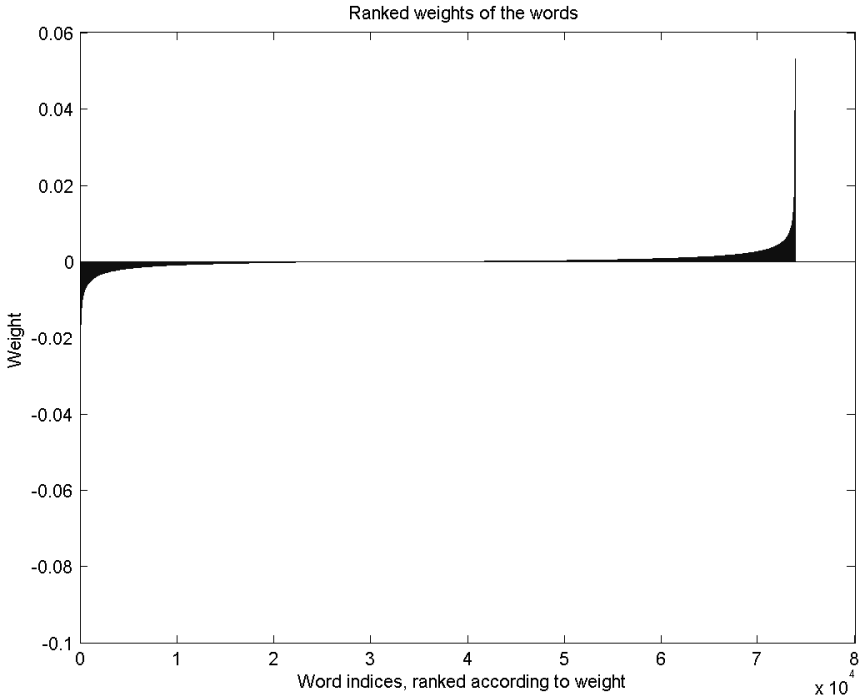

Figure 1 The size of the weights for all 73,730 words, ranked left-to-right according to increasing weights.

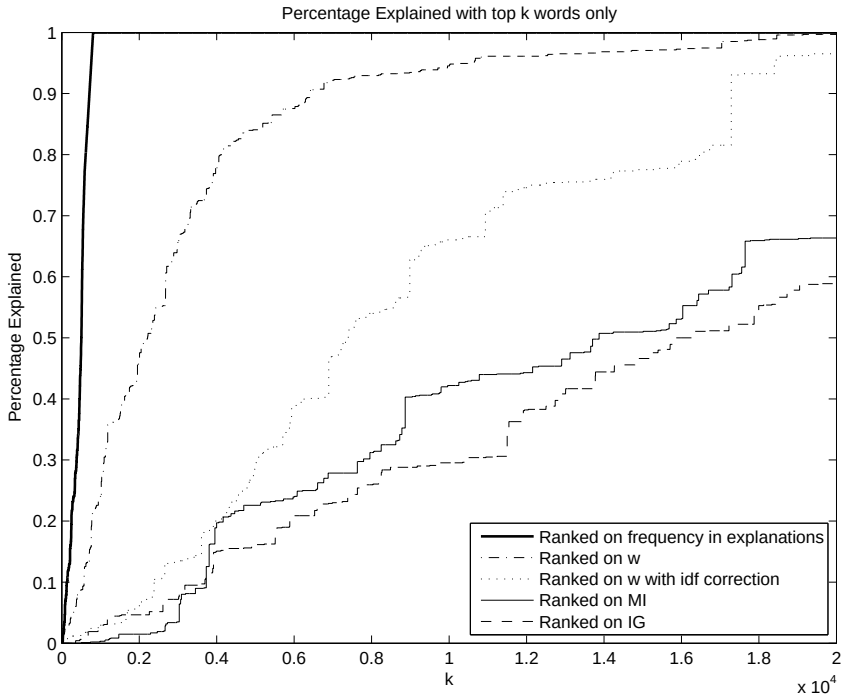

Figure 2 Percentage of 100 adult-classified test instances explained when considering only the top k words, ranked according to the frequency of occurrence in the explanations, the weights (w), the weights with idf correction, mutual information (MI) and information gain (IG).

Ranking based on				
Mutual Information	Information Gain	Size of weight	Size of weight with idf correction	Frequency of word occurring in the explanations
primarykey	privacy	welcome	permanently	adult
sessionid	policy	enter	fw	age
youtubeid	home	adult	welcome	enter
webplayerrequiredgeos	us	permanently	compuserve	site
vnesfrsgphplitgrmxnlkrause	advertise	site	copyrightc	sex
videocategoryids	about	age	prostitution	years
usergeo	adult	usc	acronym	material
latestwebplayerversion	search	searches	tribenet	are
isyoutubepermalink	comments	over	amateurbasecom	sites
isyoutube	contact	erotic	gorean	hardcore

Table 1 Global explanation of the model by listing the top words providing evidence for the adult class. Five rankings are considered: based on mutual information, information gain, weights of the words, weights of the words with idf correction (weight multiplied with idf of word), and the frequency of the word occurring in the explanations.

Ranking of some chosen intuitive words					
Word	Mutual Information	Information Gain	Size of weight	Size of weight with idf correction	Frequency of word occurring in the explanations
sex	2633	51	65	1675	5
porn	1544	86	712	4951	32
xxx	1327	143	10582	19813	558
adult	3034	7	3	48	1
prostitution	5370	5067	20	6	368
girls	916	3997	760	6135	117

Table 2 The rankings of some expert-chosen class-indicative words. When listing only the top k words, a very large k is needed before these words are included.

instance-level explanation algorithm not only for obtaining insights into the individual decisions, but also for understanding the model overall.

When we rank the words according to how often they occur in explanations, we obtain the line with the maximal area underneath. For the 100 classified instances, a total of 810 unique words are used in all the explanations (where we consider maximum 10 explanations for a single data instance). This already suggests the wide variety of words that are present in the explanations. The instance-based explanations can be aggregated to a global explanation by listing the words that occur most frequently in the explanations, as shown in the final column of Table 1—which provides yet another benefit of the instance-level explanations. We will not explore this further, as it is peripheral to the main focus of this paper.

4.1.2. Instance-level explanations None of the previously published instance-level explanation methods are able to handle many thousands of variables, so they can not be applied to this domain. We'll show now that SEDC is effective, and fast as well.

Explanation 2 shows several typical explanations for classifications of test documents. We show the first three explanations of test instances with explanations that are appropriate for publication. These explanations demonstrate several things. First, they directly address suggestion (i) just

above: in fact, documents generally do not need many, many words to be explained. They also provide evidence supporting suggestion (ii): the words in the individual explanations are quite different, including explanations in different languages.

Explanation 2: Some explanations why a web page is classified as having adult content for web pages of the test set.

Explaining document 13 (class 1) with 61 features and class 1 ...
Iteration 7 (from score 0.228905 to -0.00155753): If words (submissive pass hardcore check bondage adult ac) are removed then class changes from 1 to -1 (1 sec)
Iteration 7 (from score 0.228905 to -0.00329069): If words (submissive pass hardcore check bondage adult access) are removed then class changes from 1 to -1 (1 sec)
Iteration 7 (from score 0.228905 to -0.00182021): If words (submissive pass hardcore check bondage all adult) are removed then class changes from 1 to -1 (1 sec)

Explaining document 30 (class 1) with 89 features and class 1 ...
Iteration 4 (from score 0.894514 to -0.0108126): If words (searches nude domain adult) are removed then class changes from 1 to -1 (1 sec)
Iteration 6 (from score 0.894514 to -0.000234276): If words (searches men lesbian domain and adult) are removed then class changes from 1 to -1 (1 sec)
Iteration 6 (from score 0.894514 to -0.00225592): If words (searches men lesbian domain appraisal adult) are removed then class changes from 1 to -1 (1 sec)

Explaining document 32 (class 1) with 51 features and class 1 ...
Iteration 8 (from score 0.803053 to -0.0153803): If words (viejas sitios sexo mujeres maduras gratis desnudas de) are removed then class changes from 1 to -1 (1 sec)
Translation: old mature women sex sites free naked of
Iteration 9 (from score 0.803053 to -7.04005e-005): If words (viejas sitios mujeres maduras gratis desnudas de contiene abuelas) are removed then class changes from 1 to -1 (1 sec)
Translation: old mature women free sites containing nude grandmothers
Iteration 9 (from score 0.803053 to -0.00304367): If words (viejas sitios mujeres maduras gratis desnudas de contiene adicto) are removed then class changes from 1 to -1 (1 sec)
Translation: old sites free naked mature women contains addict

Explaining document 35 (class 1) with 36 features and class 1 ...
Iteration 6 (from score 1.04836 to -0.00848977): If words (welcome fiction erotic enter bdsm adult) are removed then class changes from 1 to -1 (0 sec)
Iteration 6 (from score 1.04836 to -0.10084): If words (welcome fiction erotica erotic bdsm adult) are removed then class changes from 1 to -1 (1 sec)
Iteration 6 (from score 1.04836 to -0.0649064): If words (welcome kinky fiction erotic bdsm adult) are removed then class changes from 1 to -1 (1 sec)

We can examine the size of explanations more systematically by referring to the explanation performance metrics introduced in Section 3.1. The top-left plot in Figure 3 shows the percentage of the test cases explained (PE) when an explanation is limited to a maximum number of words (on the horizontal axis). We see that almost all the documents have an explanation comprising fewer than three dozen words, and more than half have an explanation with fewer than two dozen words. Figure 3 also shows that, not too surprisingly, the number of words in the smallest explanation

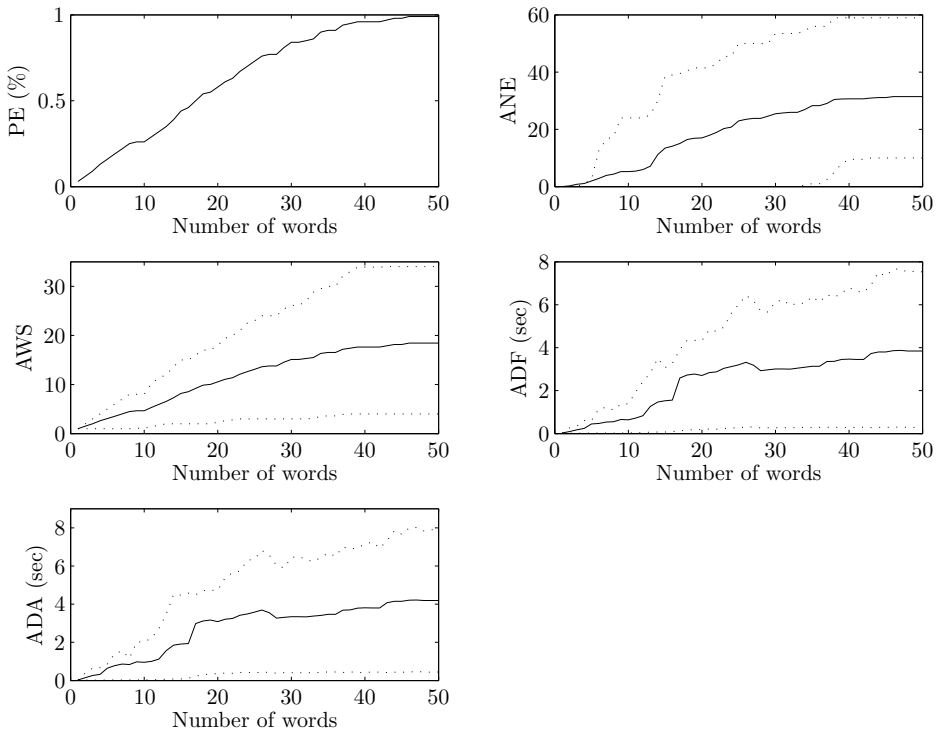

Figure 3 Explanation performance metrics in terms of maximal number of words allowed in an explanation. Both the performance and the complexity increase with the number of words. Shown are percentage explained (PE), average number of explanations given (ANE), average number of words in the smallest explanation (AWS), average duration to find the first explanation (ADF) and average duration to find all explanations (ADA). Next to the average metrics, the 10th and 90th percentiles are also shown (dashed lines).

(AWS plot) and the number of explanations (ANE plot) both grow as we allow larger and larger explanations.[10]

More interestingly, examining these performance metrics gives insight into how the classification model is functioning in this application domain. Specifically, the plots show that document explanation sizes vary quite smoothly and that there seem to be many different explanations for documents. The former observation suggests that the strength of the individual evidence varies widely: some cases are classified by aggregating many weak pieces of evidence, others by a few strong pieces of evidence (and some, presumably by a combination of strong and weak). The latter observation suggests substantial redundancy in the evidence available for classification in this application. Figure 3 also shows that for this particular problem, explanations can be produced fairly quickly using SEDC. This problem is of moderate size; real-world document classification problems can be much larger, in terms of documents for training, documents to be classified, and the vocabulary. Therefore, a word about scaling up is in order.

[10] In the experiments, we limit ourselves to searching for 10 explanations: if 10 or more explanations have been found, no further word expansions/iterations are attempted.

Let us first consider a linear model. For a document with m_D unique words, SEDC evaluates sequentially m_D "documents" (each the original document with 1 word removed), then iteratively works on the best of these leading to the evaluation of $m_D - 1$ documents (each the original with 2 words removed); next $m_D - 2$ documents are evaluated, and so on. When an explanation of size s is found a total of $O(s \times m_D)$ evaluations have occurred. The computational complexity depends therefore on (1) the time needed for a model evaluation, and text classifiers can be very fast, (2) the number of words needed for an explanation s, which in our case study went to about 40, and (3) the number of unique words in the document m_D, which is generally very small as compared to the overall vocabulary. Most importantly, the computational complexity is independent of the overall size of the vocabulary, unlike previous instance-level explanation approaches. This complexity could be lowered further for linear models to $O(s)$ by incrementally evaluating the word combinations with the next-most-highly-ranked word removed (recall Lemma 1 and Theorem 1). Our implementation does not include this speed-up mechanism as we wish to present a technique applicable to all models and not just to linear ones.

For a non-linear model, some backtracking will likely also occur, when a local minimum has been found, and thus removing any other word leads the score to increase again. The extent to which this occurs depends on the shape of the model's decision boundary. Considering word combinations of two words, backtracking once will lead to $m_D + 2 \times m_D$ evaluations instead of $m_D + m_D$. Worst case scenario, backtracking over all words occurs, leading to $m_D + m_D^{m_D}$ evaluations. Thus, the worst case complexity grows exponentially with the depth of the search tree. However, as we will show in the subsequent experiments, the heuristic approach is quite fast for the tasks to which we have applied it, and is able to provide explanations in a matter of seconds for the non-linear SVM technique with a radial basis function (RBF) kernel (a popular non-linear model). Importantly, once again, the complexity is independent on the size of the vocabulary.

Finally, recall that these experiments were conducted on a desktop PC. Further speed improvements could easily be obtained with the high-performance computing systems typically used by organizations that build text classifiers from massive data.

4.2. Hyper-explanations

Conducting this case study brought to the fore some additional issues regarding explaining documents classifications—issues that (at least for us) needed to be clarified carefully. Specifically, a procedure for producing explanations of document classifications may provide no explanation at all. Why not? A document's explanation may be non-intuitive. Then what? There are several classes of reasons for these behaviors, which we group into *hyper-explanations*.

4.2.1. Hyper-explanations for the lack of an explanation We distinguish between cases where the predicted class is the default class (hyper-explanations 1), and those where the predicted class is the non-default class (hyper-explanation 2).

Hyper-explanation 1a: no evidence present. The default class is predicted and no evidence for either class is present. For example, this would be the case when all words in the document have zero weights in the model or no words present are actually used in the model.

Technically, this case falls outside the scope of this paper's development, since we are specifically considering explaining why a document is classified as a non-default class. Nevertheless, this may be a practically important situation that cannot simply be ignored. For example, this case may have been brought to a manager's or developer's attention as a "false negative error"—i.e., it should have been classified as a positive example. In this case the hyper-explanation explains exactly why the case was classified as being negative—there was no model-relevant evidence—and can be a solid starting point for a management/technical discussion about what to do about it. For example, it may be clear that the model's vocabulary needs to be extended.

Hyper-explanation 1b: no evidence of non-default class present. The default class is predicted and only evidence in support of the default class is present. This is a minor variation to Hyper-explanation 1a, and the discussion above applies regarding explaining false negatives and providing a starting point for discussions of corrective actions.

Hyper-explanation 1c: evidence for default class outweighs evidence for the non-default class. A more interesting and complex situation is when, in weighing evidence, the model's decision simply comes out on the side of the default class. In this case an immediate reaction may be to apply the explanation procedure to generate explanations of why the case was classified as being default (i.e., if these words were removed, the class would change to positive). However, when the case truly is of the "uninteresting" class, the explanations returned would likely be fairly meaningless, e.g., "if you remove all the content words on the page except the bad words, the classifier would classify the page as a bad page." However, applying the procedure may be very helpful for explaining false negatives, because it would show the words that the model feels trump the positive-class-indicative words on the page (e.g., if you remove the medical terminology on the page, the classifier would rate the page as being adult). This again could provide a solid foundation for the process of improving the classifiers.

Within our safe advertizing application, an explanation for all 46 false negatives is found, indicating that indeed adult words are present but these are outweighed by the non-adult, negative words. Example explanations of such false negatives are given in Explanation 3. For some words like 'blog' it seems logical to have received a large non-adult/negative weight. The word 'bikini'

seemingly ought to receive a non-adult weight as well, as swimsuit sites are generally not considered to be adult content by raters. However, some pages mix nudes with celebrities in bikinis (for example). If not enough of these are in the training set, it potentially would cause 'bikini' to lead to a false negative. Many other words however can be found in the explanations that do seem to be adult-related (such as 'handjobs'), and as such should receive a positive weight. All the words are great candidates for human feedback to indicate which of these words actually are adult related and potentially update the model's weights (a mechanism known as active feature labeling (Sindhwani and Melville 2008)) or review the labeling quality of the web pages with the word. The words occurring most in these explanations of false negatives (when considering only the first explanation) are 'found', 'blog' and 'policy'. The seemingly-adult related words are not found when examining the words with most negative weights, again supporting the need to look at explanations separately, on an instance level.

Explanation 3: Explanations of web pages misclassified as non-adult (false negatives), which indicate which words the model feels trump the positive-class-indicative words.

Explaining document 10 (class 1) with 31 features and class -1 (score -0.126867)...
Iteration 4 (from score -0.126867 to 0.00460739): If words (policy gear found blog) are removed then class changes from -1 to 1 (0 sec)

Explaining document 13 (class 1) with 50 features and class -1 (score -0.123585)...
Iteration 4 (from score -0.123585 to 0.000689515): If words (sorry miscellaneous found about) are removed then class changes from -1 to 1 (0 sec)

Explaining document 11 (class 1) with 198 features and class -1 (score -0.142504)...
Iteration 2 (from score -0.142504 to 0.00313354): If words (watch bikini) are removed then class changes from -1 to 1 (1 sec)

Explaining document 31 (class 1) with 22 features and class -1 (score -0.0507037)...
Iteration 4 (from score -0.0507037 to 0.00396628): If words (search handjobs bonus big) are removed then class changes from -1 to 1 (0 sec)

Hyper-explanation 2: too much evidence of non-default class present. No explanation is provided because, although a non-default class is predicted, there are so many words in support of this class that one needs to remove almost all of them before the class will change. The situations when this will occur fall along a spectrum between two fundamentally different reasons:

1. There are very many words each providing weak evidence in support of the class. Thus, the explanation exceeds the bound given to the algorithm, or the algorithm does not return a result in a timely fashion. In Figure 4, the (middle) line for the explanation with the most words shows that if the number of allowed words is below 40, no explanation is found. This lack of explanation can be explained by this hyper-explanation, as too many adult-related words are present for a short explanation to be found.

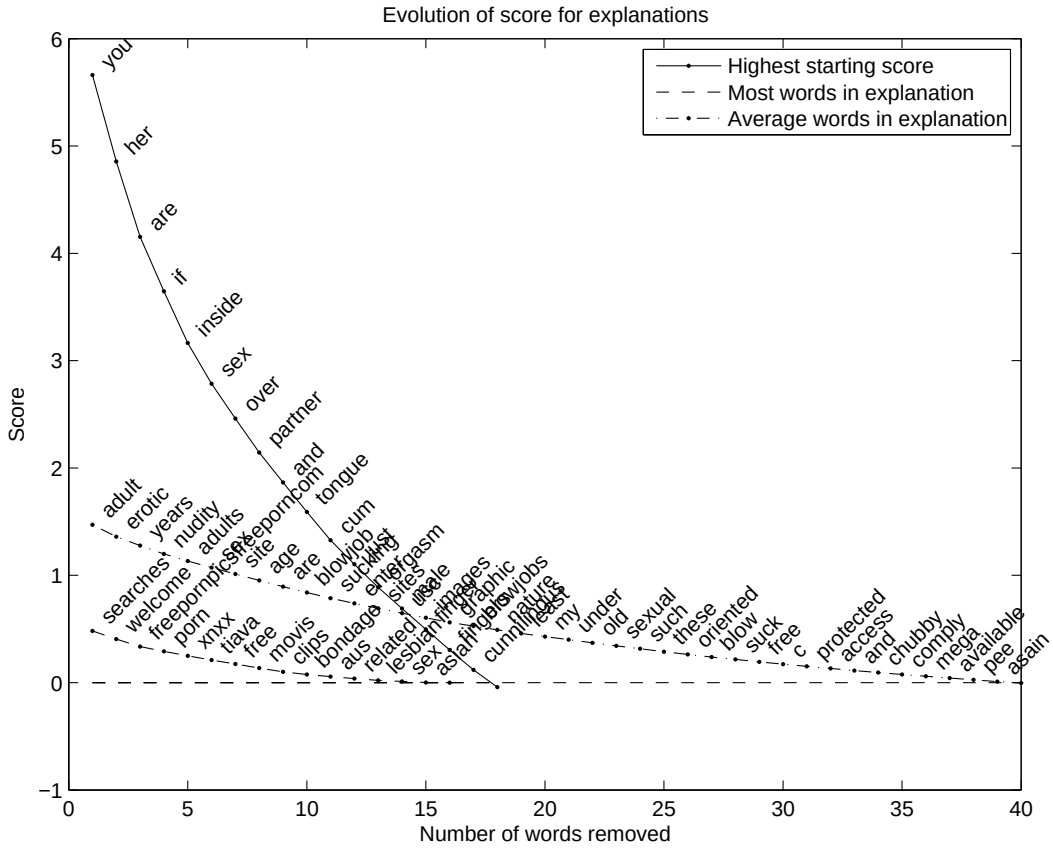

Figure 4 Score evolution when removing words from the three selected documents: the one with highest starting score, the one with the most words in an explanation and a document with average number of words in an explanation. The class changes to non-adult when the score falls below zero.

2. There are very many words each providing strong evidence. In this case, the procedure may not be able to get the score below the threshold with a small explanation—because there is just so much evidence for the class. The full upper line with the highest starting score in Figure 4 shows such an example: when allowing fewer than 15 words in an explanation, the score remains above the threshold and no explanation can be given.

This lack of base-level explanation can be mitigated (partially) by presenting "the best" partial explanation as the search advances.

4.2.2. Hyper-explanations for non-intuitive explanations Explanations are always correct in the technical sense—removing the words by definition changes the class. However, it is possible that the explanation clashes with the user's intuition. Several reasons exist for this:

The data instance is misclassified.

The explanations of some of the web pages that are misclassified by the SVM model are listed in Explanation 4 (only the first explanation is shown). For these pages the predicted class is adult,

while the human-provided class label is non-adult (false positives). These three explanations indicate strongly that the web pages actually contain adult content and the human-provided label seems wrong. On the other hand, in other cases, explanations indicate that their web pages seem to be non-adult and hence are probably misclassified. Examples are given in Explanation 5.[11] Such explanations provide very useful support for interactive model development, as the technical/business team can fix training data or incorporate background knowledge to counter the misclassification.

The data instance is correctly classified, but the explanation just does not make sense to the business users/developers.

This case is particularly problematic for any automated explanation procedure, since providing explanations that "make sense" requires somehow codifying in an operationally useful way the background knowledge of the domain, as well as common sense, which to our knowledge is (far) beyond current capabilities (and certainly beyond the scope of this paper). Nevertheless, we still can provide a quite useful hyper-explanation in the specific and common setting where the document classification model had been built from a training set of labeled instances (as in our case study). Specifically:

Hyper-explanation 3: Show similar training instance. For a case with a counter-intuitive explanation, we can show "similar" training instances with the same class. The similarity metric in principle should roughly match that used by the induction technique that produced the classifier. Such a nearest-neighbor approach can provide insight in two ways. (1) If the training classifications of the similar examples do make sense, then the user can understand why the focal example was classified as it was. (2) If the training classifications do not make sense (e.g., they are wrong), then this hyper-explanation provides precise guidance to the data science team for improving the training,[12] and thereby the model.

Consider for example document 8. Explanation 5 suggests strongly that it contains non-adult content, even though the model classifies it as adult. The web page most similar to document 8 is also classified as adult and has 44 (out of 57) words which are the same, which are listed in Explanation 6. This is a web page with a variety of topics, and probably a listing of links to other websites. This sort of web page needs further, expert investigation for use in training (and evaluating) models for safe advertising. It could be that labelers have not properly examined the

[11] Our models are limited by the data set obtained for the case study. By our understanding, models built for this application from orders-of-magnitude larger data sets are considerably more accurate; nonetheless, they still make both false-positive and false-negative errors, and the general principles illustrated here apply.

[12] Data cleaning is a very important aspect of the data mining process that has received relatively little treatment in the research literature. One of the main data cleaning activities in classifier induction is "fixing" labels on mislabeled training data.

entire web site; it may be that there indeed is adult content in images that our text-based analysis does not consider; it may be that these sites simply are misclassified, or it may be that in order to classify such pages correctly, the data science team needs to construct specifically tailored feature to deal with the ambiguity.

Explanation 4: Explanations of web pages misclassified as adult (false positives), which indicate that the model is right and the class should have been adult (class 1).

Explaining document 1 (class -1) with 180 features and class 1 (score 1.50123)...
Iteration 35 (from score 1.50123 to -0.00308141): If words (you years web warning usc these sites site sexual sex section porn over offended nudity nude models material male links if hosting hardcore gay free explicit exit enter contains comic club are age adults adult) are removed then class changes from 1 to -1 (53 sec)

Explaining document 2 (class -1) with 106 features and class 1 (score 0.811327)...
Iteration 24 (from score 0.811327 to -0.00127533): If words (you web warning under und these site porn over offended nude nature material links illegal if here exit enter blonde are age adults adult) are removed then class changes from 1 to -1 (15 sec)

Explaining document 3 (class -1) with 281 features and class 1 (score 0.644614)...
Iteration 15 (from score 0.644614 to -0.00131314): If words (you sex prostitution over massage inside hundreds here girls click breasts bar) are removed then class changes from 1 to -1 (29 sec)

Explanation 5: Explanations of truly misclassified web pages (false positives).

Explaining document 8 (class -1) with 57 features and class 1 (score 0.467374)...
Iteration 7 (from score 0.467374 to -0.0021664): If words (welcome searches jpg investments index fund domain) are removed then class changes from 1 to -1 (3 sec)

Explaining document 16 (class -1) with 101 features and class 1 (score 0.409314)...
Iteration 8 (from score 0.409314 to -0.000867436): If words (welcome und sites searches domain de b airline) are removed then class changes from 1 to -1 (5 sec)

Explaining document 32 (class -1) with 66 features and class 1 (score 0.124456)...
Iteration 2 (from score 0.124456 to -0.00837441): If words (searches airline) are removed then class changes from 1 to -1 (0 sec)

Explanation 6: Hyper-explanation 3 showing the words of the web page most similar to document 8. This most similar web page is classified as adult, providing a hyper-explanation of why document 8 is also classified (incorrectly) as adult.

and, articles, at, buy, capital, check, china, commitment, dat, file, files, for, free, fund, funds, high, hot, in, index, instructionalwwwehowcom, international, internet, investing, investment, investments, jpg, listings, mutual, out, performance, project, related, results, return, searches, social, sponsored, temporary, tiff, to, trading, vietnam, web, welcome.

4.3. News Item Categorization

To demonstrate generality and to illustrate some additional properties of the method we now apply the explanation method to a second domain: classifying news stories. The 20 Newsgroups data set is a benchmark data set used in document classification research. It contains about 20,000 news items partitioned evenly over 20 newsgroups of different topics, and has a vocabulary of 26,214 different words (Lang 1995). The 20 topics can be categorized into seven top-level usenet categories with related news items: alternative (alt), computers (comp), miscellaneous (misc), recreation (rec), science (sci), society (soc) and talk (talk). One typical problem addressed with this data set is to build classifiers to identify stories from these seven high-level news categories—which for our purposes gives a wide variety of different topics across which to provide document classification explanations. Looking at the seven high-level categories also provides realistic richness to the task: in many real document classification tasks, the class of interest is actually a collection (disjunction) of related concepts (consider, for example, "hate speech" in the safe-advertising domain).

We build a classifier system to distinguish the seven top-level categories using all words in the vocabulary. This permits us to examine a wide variety of explanations of different combinations of true class and predicted class, in a complicated domain—but one where we have at least a high-level intuitive understanding of the classes. The examination shows that even for news items grouped within the same top-level category, the explanations for their classifications can vary greatly and are intuitively related to their true lower-level newsgroup.

4.4. Results

The classifier system for distinguishing the seven top-level newsgroups (alt, comp, misc, rec, sci, soc, talk) operates in a one-versus-others setup—i.e., seven classifiers are built, each distinguishing one newsgroup from the rest. For training (on 60% of the data) and for prediction (remaining 40% as test data), if a news item is (predicted to be) from the given newsgroup, the class variable is set to one; if not the class variable is set to zero. To demonstrate the method with different types of model, here we build both linear and non-linear SVM classifiers. The non-linear SVM is built with the LIBSVM package (Chang and Lin 2001) and uses a radial basis function (RBF) kernel with hyperparameters tuned using a grid search.

In Table 3, each cell shows at least one explanation (where possible) of an example from one of the 20 low-level categories (specified in the row header) being classified into one of the top-level categories (specified in the column header). If no explanation is given in a cell, either no misclassified instances exist, which occurs most, or no explanation was found with maximum 10 words. The shaded cells on the diagonal are the explanations for correct classifications; the rest are explanations for errors. For example, the first explanation in the upper-left cell (excluding the

header rows) shows that this correct classification of a news story in the alt.atheism category is explained by the inclusion of the terms 'ico', 'bibl', 'moral', 'god' and 'believ'–if these words alone are removed, the classifier would no longer place this story correctly into the alt category.

Several cells below we see explanations for why a sci.med story was misclassified as belonging to alt: because of the occurrence of the word 'atheist' (first explanation), or the words 'god' and 'believe' (second explanation). Further investigation of this news story reveals it concerns organ donation. More generally, the explanations shown in Table 3—the correctly classified test instances (grayed cells on the diagonal)—usually are indeed intuitively related to the topic.

The categories themselves often occur as words in the explanations, such as 'hardwar', 'microsoft', 'mac' and 'space'. Importantly, the different subcategories of the newsgroups show different explanations, which motivates using instance- rather than global-level explanations. For example, for the computer newsgroup (shown in the second column), the terms used to explain classifications from the different subgroups are quite different and intuitively related to the specific subgroups.

The misclassified explanations (outside of the shaded cells) often show the ambiguity of certain words as reason for the misclassification. For example 'window' is a word that can be related to computer, but also can be seen as words related to automobiles. The explanations for the misc.forsale news items indicate they are most often misclassified because the item that is being sold comes from or is related to the category it is misclassified in. With this individual-instance approach, similar ambiguities as well as intuitive explanations for each of the subgroups also can be found for the other categories. The results also demonstrate how the explanations can hone in on possible overfitting, such as with 'unm' and 'umd' in the cells adjacent to the upper-left cell we discussed above.

The explainability metrics when allowing a maximum of 10 words in an explanation are shown in Table 4. Although a high percentage of the test instances is explained (PE around 90-95% for all models) still some instances remain unexplained. If we allow up to 30 words in an explanation, all instances are explained for each of the models. Of particular note is that for this widely used benchmark with a vocabulary of 26,214 words, on average only a small fraction of a second (ADF of 0.02-0.08 seconds) is needed to find a first explanation. As previously mentioned, this is because our SEDC explanation algorithm is independent of the vocabulary size. Explaining the non-linear model requires more time, since backtracking occurs and the model evaluation takes longer than for a linear model. Nevertheless, on average still less than a second is needed to find an explanation.

	Classification models in one-versus-others setup: 'newsgroup' versus not 'newsgroup' Explanations why news items are classified as 'newsgroup'			
	alt vs not alt	**comp vs not comp**	**misc vs not misc**	**rec vs not rec**
alt.atheism	ico bibl moral god believ ico bibl moral god read ico bibl moral accept god	unm carina screen carina join	wustl distribut wustl 5 wustl origin	com univers distribut
comp.graphics	umd wam mistak cant	quicktim 3do centris resolut card program quicktim 3do centris resolut ac card quicktim 3do centris resolut fax card	bigwpi wpi distribut bigwpi wpi pleas bigwpi wpi email	nb canada ca nb luck canada nb archiv canada
comp.os.ms-windows.misc		mous microsoft cant mous microsoft solution mous microsoft switch	distribut look pleas	6 tom archiv com
comp.sys.ibm.pc.hardware		hardwar thank hardwar appreci adam hardwar	distribut repli call	cornel buffalo buffalo cc wonder ubvmsb buffalo cc
comp.sys.mac.hardware	kmr4po read kmr4po follow kmr4po note	vga monitor mac advenc card am vga monitor mac advenc card repli vga monitor mac advenc card thank	offer sale distribut offer sale card jame offer sale	univers recent price
comp.windows.x		enterpoop lcs fax enterpoop lcs mit enterpoop xpertexpo lcs inc	pleas includ send	street final list 2154 street final com 2154 street final pleas
misc.forsale		driver program driver card pc driver	sale 2190 pc mention	insur gasket massachusett ser gasket jacket massachusett
rec.autos		window call window email window 4	distribut 3 compani	geico insur distribut geico insur ca geico insur usa
rec.motorcycles		greyscal color greyscal pictur greyscal directori	mile pad rosevil deal	dod ottawa ca ottawa canada
rec.sport.baseball			offer game 3 game 5	miller brave gatech nl seri team technologi game miller brave gatech nl seri team institut game miller brave gatech nl seri team plai game
rec.sport.hockey		michel comput michel 4 co michel	susan game call buffalo game	buffalo ny team bruin buffalo team sabr buffalo team
sci.crypt	mathew rusnew mantis umd consult couldnt agre rusnew mantis umd consult couldnt stop	42 print messag 42 print seen 42 print net	ohio cincinnati victor	usa list free
sci.electronics		softwar prefer appl	sell price email pleas sell price game email ncsu sell price email	univers distribut ca
sci.med	atheist god believ god start	lcs mit address thank lcs laboratori mit address lcs mit address email am	nyx denver du denver dept distribut	canada cc bad pleas univers canada cc bad pleas thank canada cc bad i'v pleas
sci.space		michel help site help help thank am	internet servic institut	riversid due riversid ucr riversid prbaccess com
soc.religion.christian	atheist	wrote technologi 9	call person includ	chanc dave princeton
talk.politics.guns		richard drive richard fax bryan richard	holonet norton internet holonet norton modem holonet norton pete	sfasu arlen thank arlen pleas
talk.politics.mideast	wrote evid religion	ai repli ai mit ai cant 3	hous amherst pl7	cc columbia lion
talk.politics.misc	religi god religi religion islam religi	cwru jone cleveland western	ohio jone hela ins cleveland reserv western usa 2	car watch jm
talk.religion.misc	bill explain cration	site ca system usa system	institut gold polytechn	refer mike univ

	Classification models in one-versus-others setup: 'newsgroup' versus not 'newsgroup' Explanations why news items are classified as 'newsgroup'		
	sci vs not sci	**soc vs not soc**	**talk vs not talk**
alt.atheism	latech scisur rayengr help	translat familiar translat god	ha atom 2000 moral object evid ha overwhelm atom 2000 moral object microscop ha atom 2000 moral object
comp.graphics	map pub inc pub ftp	scott pleas scott read scott answer	david happen list
comp.os.ms-windows.misc	public date std	book pa steven	speak limit stand
comp.sys.ibm.pc.hardware	nz mark nz 1.1 nz network		address student utexa
comp.sys.mac.hardware	bounc suppli bounc circuit sync bounc happen		purdu cc center pure cc
comp.windows.x	nz aukuni time aukuni scienc	scienc sorc upenn	re time name
misc.forsale	tube catalog umb etc	pa sex accept sex hell	usa 21 gun
rec.autos	max low fone max cycl fone max pl9 effect fone	chuck discuss pleas discuss read	utexa call utexa center utexa care
rec.motorcycles	ibm week fone rochest fone 10		righteous racist stupid mean righteous racist stupid own righteous racist stupid opinion
rec.sport.baseball	list 10 list scienc std list	dt nswc carderock	buffalo love cc buffalo stand cc buffalo stori cc
rec.sport.hockey	ericsson inc ericsson commun ericsson user	oppos csd chuck	john boulder center boulder depart
sci.crypt	inform commun offic		congress law john preced congress john nagl congress john
sci.electronics	adcom preamp chip sound preamp network chip	god accept recent	re david citi
sci.med	handed rsilverworld sight domin eye commun handed rsilverworld sight domin eye indic handed rsilverworld sight domin guest eye look	sex grade fysic fysic speak reason	perot 16 happen edward happen
sci.space	space nasa follow nasa scienc	book discuss fysic	terror moral govern terror moral law terror moral major
soc.religion.christian	greet marie angel gabriel greet mari 12 gabriel greet mari various	religion pleas religion question religion follow	homosexu abus behavior love abus sexual love peopl
talk.politics.guns	chip explode medic understand	marri christ life marri christ view marri christ religion	batf waco clinton question batf waco clinton law batf waco clinton evid
talk.politics.mideast	ai amend lab amend messag 10	ab4zvirginia beyer ab4zvirginia beyer andi blanket ab4zvirginia beyer andi	holocaust arab militari plan evid kill holocaust arab militari attack evid kill holocaust arab militari reach evid kill
talk.politics.misc	acid scienc acid commun acid sorc	serbian bomb york 2 bomb york position	homosexu moral law homosexu moral stop homosexu moral pass
talk.religion.misc	messag institut apr	pa christian mormon faith christian 2 mormon faith hous christian	malcolm weapon jew christian malcolm weapon jew kill malcolm weapon jew hous

Table 3 Explanations are shown why documents from the newsgroup shown at the beginning of the row are classified in the newsgroup shown at the top of the column.

Model	Linear SVM						Non-linear RBF SVM					
	PCC	PE	ANE	AWS	ADF	ADA	PCC	PE	ANE	AWS	ADF	ADA
alt	81.5%	96.1%	18.5	2.7	0.05	0.16	76.8%	95.7%	30.1	2.5	0.62	1.35
comp	93.7%	89.1%	13.3	3.1	0.05	0.12	94.9%	81.7%	12.4	3.3	0.54	0.88
misc	92.8%	98.1%	12.9	1.9	0.02	0.12	90.5%	96.6%	17.0	1.8	0.14	0.38
rec	94.2%	94.8%	13.7	2.4	0.04	0.11	93.6%	92.9%	16.7	2.4	0.40	0.79
sci	85.4%	93.5%	19.6	2.7	0.06	0.15	83.1%	90.4%	23.16	2.7	1.01	1.62
soc	94.2%	94.4%	16.9	1.8	0.03	0.15	90.2%	91.5%	29.5	2.4	0.39	0.79
talk	88.5%	92.1%	23.8	2.5	0.08	0.21	86.8%	90.0%	28.5	2.0	1.3	2.9

Table 4 Explanation performance metrics on the test set of the 20 newsgroups data set for a linear (left) and non-linear (right) SVM model and explanations of maximum 10 words.

These results in a second domain, with a wide range of document topics, provide support that our general notion of instance-level document classification can provide important insight into the functioning of text classifiers, and that the SEDC method is generally effective and pretty fast as well. Further, this second study provides a further demonstration of the futility of global explanations in domains such as this: there are so many different reasons for different classifications. At best they would be muddled in any global explanation, and likely they would simply be incomprehensible.

5. Discussion, Limitations and Conclusions

In this paper, we followed the guidelines set forth by Hevner et al. (2004) for designing, executing and evaluating research within design science to explain documents' classifications. The business problem we address is obtaining insight into a document classification model such that (1) the manager using it understands how decisions are being made, (2) the customers affected by the decisions can be advised why a certain action regarding them is taken, and (3) the data science/development team can improve the model iteratively. Further, (4) document classification explanations can provide insight into the business domain.

We found that global explanations in the form of a decision tree or a list of the most indicative words do not provide a satisfactory solution. Moreover, previously proposed explanation methods on the data-instance level all define explanations as real-valued vectors of the same size as the input space. Given the huge dimensionality of document classification problems, these techniques also do not provide a solution to the business problems. Thus, a new approach is needed. With the technical constraints of high-dimensional data in mind, we addressed this business problem by creating an explanation as a "necessary" set of words—a minimal set such that after removal the current classification would no longer be made. We presented a search algorithm (SEDC) for finding such explanations—the algorithm is optimal for linear binary-classification models, and heuristic for non-linear models. We introduced important aspects of the evaluation of such a system, and then provided empirical evaluations of the performance of the algorithm on two different document

classification domains. The evaluations show that SEDC is able to provide these explanations in a matter of seconds.

In terms of effectiveness, the results show that the explanations are quite comprehensible, comprising a few to a few dozen words. The words in the explanations vary greatly across the explanations, even with words in different languages, which supports the claim that existing global explanations are inadequate for such document classification domains. We see very different explanations for different cases. These results suggest a different route for producing global explanation models for document classification. Rather than trying to produce a small, high-fidelity replica (as with prior approaches), instead produce a *large* high-fidelity replica, that captures all the different sorts of classifications the model is making. This may sound counter-intuitive, since in prior work model size often is equated with comprehensibility. However, a model that comprises a large number of individually comprehensible subcomponents (e.g., a large set of small rules) may provide useful insight. Nevertheless, it would not substitute for instance-level explanations for the business problems we address.

An unexpected (to us) result of the case study was the need for various sorts of hyper-explanations. Several of these are the result of the document classification models being statistical models learned from data, and thus are subject to the main challenges of machine learning: overfitting, underfitting, and errors in the data. When classification errors are introduced due to these pathologies, even instance-level explanations may be inadequate (e.g., missing) or unintuitive. Hyperexplanations are needed for deep understanding—for example, showing training cases that likely led to the current model behavior.

As discussed in the introduction, we believe that instance-level explanation methods such as SEDC can have a substantial impact in improving the process of building document classification models. The field needs more research addressing support for the process of building acceptable models, especially in business situations where various parties must be satisfied with the results. Indeed, recent developments in machine learning and data mining arguably have moved us further away from the needed transparency, with the strong research emphasis on and seeming success of techniques resulting in complex models, such as boosting, non-linear SVMs, feature hashing (see below), etc. Managers and developers need to be able to interact to agree that a classification system is behaving appropriately.

More specifically, systems like SEDC may become a critical component of the iterative process for improving document classification models. As the case study and the news-group study showed, SEDC can identify data quality issues and model deficiencies. These deficiencies can be resolved via various mechanisms, leading to improved models directly or, alternatively, to improved data quality, which ultimately should lead to better model performance and decision making.

We hope that this new sort of instance-level explanation for document classification will provide an immediately useful method across a wide variety of business (and scientific, medical, and legal) applications where document classifications are critical. We also hope we have made the case that thinking about explanations in this way opens up a large number of new research problems and opportunities for improving the state of the art in building and using data-driven document classification systems.

References

Aggarwal, C.C., C. Chen, J.W. Han. 2010. The inverse classification problem. *Journal of Computer Science and Technology* **25**(3) 458–468.

Attenberg, J., K. Q. Weinberger, A. Smola, A. Dasgupta, M. Zinkevich. 2009. Collaborative email-spam filtering with the hashing-trick. *Sixth Conference on Email and Anti-Spam (CEAS)*.

Baehrens, D., T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K.-R. Müller. 2010. How to explain individual classification decisions. *Journal of Machine Learning Research* **11** 1803–1831.

Baesens, B., T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, J. Vanthienen. 2003. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society* **54**(6) 627–635.

Barbella, D., S. Benzaid, J.M. Christensen, B. Jackson, X. V. Qin, D. R. Musicant. 2009. Understanding support vector machine classifications via a recommender system-like approach. Robert Stahlbock, Sven F. Crone, Stefan Lessmann, eds., *DMIN*. CSREA Press, 305–311.

Bishop, C.M. 1996. *Neural networks for pattern recognition*. Oxford University Press, Oxford, UK.

Chang, Chih-Chung, Chih-Jen Lin. 2001. *LIBSVM: a library for support vector machines*.

Craven, M.W., J.W. Shavlik. 1996. Extracting tree-structured representations of trained networks. D.S. Touretzky, M.C. Mozer, M.E. Hasselmo, eds., *Advances in Neural Information Processing Systems*, vol. 8. The MIT Press, 24–30.

eMarketer. April 27, 2010. Brand safety concerns hurt display ad growth. Http://www1.emarketer.com/Article.aspx?R=1007661.

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* **9** 1871–1874.

Fawcett, T., F. Provost. 1997. Adaptive fraud detection. *Data Mining and Knowledge Discovery* **1**(3) 291–316.

Fayyad, U.M., G. Piatetsky-Shapiro, P. Smyth. 1996. From data mining to knowledge discovery: An overview. *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence, 1–34.

Hettich, S., S. D. Bay. 1996. The uci kdd archive [http://kdd.ics.uci.edu].

Hevner, A. R., S. T. March, J. Park, S. Ram. 2004. Design science in information systems research. *MIS Quarterly* **28**(1) 75–106.

Hotho, A., A. Nürnberger, G. Paass. 2005. A brief survey of text mining. *LDV Forum* **20**(1) 19–62.

Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. *European Conference on Machine Learning (ECML)*. Springer, Berlin, 137–142.

Lang, Ken. 1995. Newsweeder: Learning to filter netnews. *Proceedings of the Twelfth International Conference on Machine Learning*. 331–339.

Lessmann, S., B. Baesens, C. Mues, S. Pietsch. 2008. Benchmarking classification models for software defect prediction: A proposed framework and novel findings. *IEEE Trans. Software Eng.* **34**(4) 485–496.

Mannino, M., M. Koushik. 2000. The cost-minimizing inverse classification problem: A genetic algorithm approach. *Decision Support Systems* **29** 283–300.

Martens, D., T. Van Gestel, B. Baesens. 2009. Decompositional rule extraction from support vector machines by active learning. *IEEE Transactions on Knowledge and Data Engineering* **21**(2) 178–191.

Pang, B., L. Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* **2**(1-2) 1–135.

Platt, J. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans, eds., *Advances in Large Margin Classifiers*. MIT Press, Cambridge, MA.

Qi, X., B.D. Davison. 2009. Web page classification: Features and algorithms. *ACM Computing Surveys (CSUR)* **41**(2) 1–31.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Robnik-Šikonja, M., I. Kononenko. 2008. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering* **20** 589–600.

Schapire, Robert E., Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning* **39**(2/3) 135–168.

Sheng, Victor S., Foster Provost, Panagiotis Ipeirotis. 2008. Get another label? Improving data quality and data mining using multiple, noisy labelers. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*.

Sindhwani, V., P. Melville. 2008. Document-word co-regularization for semi-supervised sentiment analysis. *ICDM*.

Vapnik, V. N. 1995. *The nature of statistical learning theory*. Springer-Verlag, New York, NY, USA.

Štrumbelj, E., I. Kononenko. 2010. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research* **11** 1–18.

Štrumbelj, E., I. Kononenko, M. Robnik-Šikonja. 2009. Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering* **68**(10) 886–904.

Acknowledgment

We extend our gratitude to AdSafe Media and Josh Attenberg for many discussions into the problem of safe advertising. This particular data set was not necessarily used in the development of any production model used for safe advertising. Foster Provost also thanks NEC for a Faculty Fellowship.

Appendix

This appendix describes in detail existing methods for explaining individual classifications, and discusses why they are not ideal or suitable for explaining document classifications. To our knowledge, the first approach to explain classifications of individual instances that is applicable to any classification model was presented by Robnik-Šikonja and Kononenko (2008). The authors present a methodology to assign scores to each of the variables that indicate to what extent they influence the data instance's classification. As such, they define an explanation as a real-valued vector $\mathbf{e}$ that denotes the contribution of each variable to the classification of the considered data instance $\mathbf{x_0}$ by classification model M (see Definition 2). The effect of each attribute of a test instance $\mathbf{x_0}$ is measured by comparing the predicted output $f(\mathbf{x_0})$ with $f(\mathbf{x_0} \backslash A_i)$, where $\mathbf{x_0} \backslash A_i$ stands for the instance without any knowledge about attribute A_i. This is implemented by replacing the actual value of A_i with all possible values for A_i and weighting each prediction by the prior probability of that value. For continuous variables, a discretization method is applied to the variable. The larger the change in predicted output, the larger the contribution of the attribute. This change in output can be measured in various ways, using simply the difference in probabilities, the information difference or the weight of evidence. The contributions provided by the previously discussed technique are very similar to the weights in a linear model, which also denote the relative importance of each variable.

Definition 2. Robnik-Šikonja and Kononenko (2008) define an explanation of the classification of model M for data instance $\mathbf{x_0}$ as an m dimensional real-valued vector:

$$E_{RS}(M, \mathbf{x_0}) = \mathbf{e} \in \mathbb{R}^{\mathbf{m}}, \text{ with } e_i = f(\mathbf{x_0}) - \mathbf{f}(\mathbf{x_0} \backslash A_i)$$

The explanation of each attribute can easily be visualized, graphically showing the magnitude and direction of the contribution of each variable. A very simple example is given for the Titanic data set where the aim is to predict whether a Titanic passenger survived. The instance with a female, adult, third-class passenger that is classified as surviving is explained by the contributions below. The fact that the passenger is female is the main contributor for the prediction, as the contributions for age and class are very small and even in the opposite direction.

- class=third, contribution = -0.344
- age=adult, contribution = -0.034
- gender=female, contribution = 1.194

This basic approach is not able to detect the case where a change in more than one variable is needed in order to obtain a change in predicted value. Štrumbelj et al. (2009) build further on this and proposes an Interactions-based Method for Explanation (IME) that is able to detect the contribution of combinations of feature values. The explanation once again is defined as a real-valued m-dimensional vector denoting variable contributions. First, a real value number is assigned to each subset of the power set of feature values. These changes are subsequently combined to form a contribution for each of the individual feature values. In order to assess the output of the model with a subset of variables, instead of weighting over all permutations of the features values, a model is built using only the variables in the subset. Although the results are interesting, they only use data sets of dimensions maximal 13.

There are two major drawbacks of this method. Firstly, the time complexity scales exponentially with the number of variables. They report that 241 seconds are needed to explain the classification of 100 test instances for the random forests model for the highest dimensional data sets (breast cancer ljubljana which has 13 features). The authors recognize the need for an approximation method. Secondly, the explanation is not very humanly understandable, as the explanation is once again a real-valued number for each feature which denotes to what extend it contributes to the class. They verify their explanations with an expert, where an expert needs to assess whether he or she agrees with the magnitude and direction of the contribution of each feature value.

A game-theoretical perspective of their method is provided by Štrumbelj and Kononenko (2010), as well as a sampling-based approximation that does not require retraining the model. On low dimensional data sets they provide results very quickly (in the order of seconds). For the data set with most features, arrhythmia (279 features), they report that it takes more than an hour to generate an explanation for a prediction of the linear Naive Bayes model. They state:

> *The explanation method is therefore less appropriate for explaining models which are built on several hundred features or more. Arguably, providing a comprehensible explanation involving a hundred or more features is a problem in its own right and even inherently transparent models become less comprehensible with such a large number of features.*

Stated within our safe advertizing application: a vector of thousands of values does not provide an answer to the question *'Why is this web page classified as containing adult content?'* This approach is therefore not suitable for document classification, and motivates the specific focus within this paper.

Baehrens et al. (2010) also define an instance level explanation as a real-valued vectors. In this case however, the vector denotes the gradient of the classification probability output in the test instance to explain, and as such defines a vector field indicating where the other classification can be found.

Definition 3. Baehrens et al. (2010) define an explanation of the classification of model M for data instance $\mathbf{x_0}$ as an m dimensional real-valued vector, obtained as the gradient of the class probability in the instance:

$$E_B(M, \mathbf{x_0}) = \mathbf{e} \in \mathbb{R}^{\mathbf{m}}, \text{ with } e_i = \nabla p(x)|_{x=x_0}$$

For SVMs it uses an approximation function (through Parzen windowing) in order to calculate the gradient. In our document classification setup, this methodology in itself does not provide an explanation in the form that is wanted as it simply will give the direction of steepest descent towards the other class. It could however serve as a basis for a heuristic explanation algorithm to guide the search towards those regions where the change in class output is the largest. The exact stepsize and the minimal set of explaining dimensions (words) still need to be determined within such an approach.

Inverse Classification. Sensitivity analysis is the study of how input changes influence the change in the output, and can be summarized by Eq. (1).

$$f(x + \Delta x) = f(x) + \Delta f \tag{1}$$

Inverse classification is closely related to sensitivity analysis and involves "*determining the minimum required change to a data point in order to reclassify it as a member of a (different) preferred class*" (Mannino and Koushik 2000). This problem is called the inverse classification problem, since the usual mapping is from a data point to a class, while here it is the other way around. Such information can be very helpful in a variety of domains: companies, and even countries, can determine what macro-economic variables should change so as to obtain a better bond, competitiveness or terrorism rating. Similarly, a financial institution can provide (more) specific reasons why a customer's application was rejected, by simply stating how the customer can change to the good class, e.g. by increasing income by a certain amount. A heuristic, genetic-algorithm based approach is used in Mannino and Koushik (2000) that uses a nearest neighbor model.

Classifications made by a SVM model are explained in Barbella et al. (2009) by determining the minimal change in all variables needed in order to achieve a point on the decision boundary. Their approach solves an optimization problem with SVM-specific constraints. A slightly different definition of inverse classification is given in Aggarwal et al. (2010), which provides values for

the undefined variables of a test instance that result in a desired class. Barbella et al. (2009) search for explanations by determining the point on the decision boundary (hence named border classification) for which the Euclidean distance to the data instance to be explained is minimal.

Definition 4. Barbella et al. (2009) implicitly define an explanation of the classification of model M for data instance $\mathbf{x_0}$ as the m dimensional real-valued input vector closest to $\mathbf{x_0}$, for which the predicted class is different from the predicted class of $\mathbf{x_0}$:

$$E_{IC}(M, \mathbf{x_0}) = \mathbf{e} \in \mathbb{R}^m = \text{argmin}_{\mathbf{e}} \sum_{j=1}^{n} (e_j - x_{0j})^2 \text{ and } f(\mathbf{e}) = 0$$

Since finding the global optimal solution is not feasible, a locally optimal solution is sought. The approach is applied on a medical data set with eight variables. The explanation provided shows a change in *all* variables. Applying this to document classification is therefore again not useful. The authors describe the appropriateness for low dimensional data only as follows:

> *our approach in the current form is most usable when the number of features of the data set is of a size that the user can eyeball all at once (perhaps 25-30 or so) (Barbella et al. 2009).*