Text Classification for Data Loss Prevention

Michael Hart[1], Pratyusa Manadhata[2], and Rob Johnson[1]

[1] Computer Science Department, Stony Brook University
{mhart,rob}@cs.stonybrook.edu
[2] HP Labs
manadhata@cmu.edu

Abstract. Businesses, governments, and individuals leak confidential information, both accidentally and maliciously, at tremendous cost in money, privacy, national security, and reputation. Several security software vendors now offer "data loss prevention" (DLP) solutions that use simple algorithms, such as keyword lists and hashing, which are too coarse to capture the features what makes sensitive documents secret. In this paper, we present automatic text classification algorithms for classifying enterprise documents as either sensitive or non-sensitive. We also introduce a novel training strategy, *supplement and adjust*, to create a classifier that has a low false discovery rate, even when presented with documents unrelated to the enterprise. We evaluated our algorithm on several corpora that we assembled from confidential documents published on WikiLeaks and other archives. Our classifier had a false negative rate of less than 3.0% and a false discovery rate of less than 1.0% on all our tests (i.e, in a real deployment, the classifier can identify more than 97% of information leaks while raising at most 1 false alarm every 100th time).

1 Introduction

Modern enterprises increasingly depend on data sharing, both inside and outside their organizations. Increased sharing has led to an increasing number of *data breaches*, i.e., malicious or inadvertent disclosures of confidential and sensitive information, such as social security numbers (SSN), medical records, trade secrets, and enterprise financial information, to unintended parties. The consequences of data breach can also be severe: violation of customers' privacy, loss of competitive advantage, loss of customers and reputation, punitive fines, and tangible monetary loss. The Ponemon Institute's 2009 *Cost of a Data Breach Study* found that a data breach costs an average of $6.6 million to an organization [26]. The Privacy Rights Clearinghouse lists almost 500 million records that have been leaked in data breaches since 2005 [11].

Security vendors have begun to offer a raft of "Data Loss Prevention" (DLP) products designed to help businesses avoid data breaches [37, 50, 53, 46, 45]. DLP systems identify confidential data on network storage servers, monitor network traffic and output channels to peripheral devices such as USB ports, and either enforce data control policies or generate reports that administrators can use to investigate potential breaches.

Although existing DLP solutions are quite sophisticated in detecting, capturing and assembling information flows, they are currently limited in their capability to recognize sensitive information. Many vendors offer solutions that rely on keywords, regular expressions and fingerprinting, but these techniques alone cannot fully capture the organization's secrets when it is re-phrased or re-formatted. More elaborate and comprehensive human annotations and access control will not solve the problem because they rely on users to encode in a machine-readable form the sensitive contents of the message. This is simply infeasible for certain types of data, too time consuming and too error prone. Security vendors now recognize[50] the need for DLP systems to learn and automatically classify sensitive materials.

In this paper we develop practical, accurate, and efficient machine learning algorithms to learn what is sensitive and classify both structured and unstructured enterprise documents as either public or private. Our scheme is practical because enterprise administrators need only provide an initial set of public and private documents. Our system trains a classifier using these documents, and then uses the resulting classifier to distinguish public and private documents. System administrators do not have to develop and maintain keyword lists, and our classifier can recognize private information, even in documents that do not have a substantial overlap with previously-observed private documents.

We summarize the results of our classifier on 5 testing corpora in Section 5 and compare the results with a baseline off-the-shelf classifier (Section 3). Our classifier achieves an average false positive rate of 0.46% and an average false negative rate of 1.6% on our testing corpora. The classifier also achieves a much lower false discovery rate (FDR), i.e., the percentage of false alarms defined as:

$$FDR = \frac{FP}{TP + FP}$$

raised by the classifier, than the baseline classifier. A low FDR is essential since users will ignore a system that frequently raises false alarms. If we assume a typical enterprise network (Section 5), then our classifier has an average FDR rate of 0.47% compared to the baseline classifier's average FDR rate of 16.65%. These results demonstrate that our classifier can meet the demanding needs of enterprise administrators.

In summary, this paper makes the following key contributions to the field of enterprise data loss prevention.

– We demonstrate that simply training a classifier combining enterprise data, both public and private, yields prohibitively high false positive rates on non-enterprise data, indicating that it will not perform well in real networks.
– We present a new algorithm for classifying sensitive enterprise documents with low false negative rates and false positive rates. This algorithm employs a new training technique, supplement and adjust, to better distinguish between sensitive, public and non-enterprise documents. Our algorithm scales to real time enterprise network traffic and does not rely on any metadata.
– We construct the first publicly available corpora for evaluating DLP systems.

The rest of the paper is organized as follows. We briefly describe a typical DLP system in Section 2 and discuss how our classifier fits into a DLP system. We introduce our classification algorithms in Section 3 and describe our test corpora in Section 4. We discuss our classification results in Sections 5 and 6. In Section 7, we compare our work with related work. We conclude with a summary and possible avenues of future work in Section 8.

2 Data Loss Prevention Systems

In this section, we describe a typical DLP system's building blocks and discuss how our proposed approach fits into the system. A DLP system aims to protect three types of data in an enterprise: *data-at-rest*, *data-in-motion*, and *data-in-use*. Data-at-rest is static data stored on enterprise devices such as document management systems, email servers, file servers, networked-attached storage, personal computers, and storage area networks. Data-in-motion is enterprise data contained in outbound network traffic such as emails, instant messages, and web traffic. Data-in-use is data being "used" by the enterprise's employees on end point devices, e.g., a file being copied to a USB drive.

Let us consider the definition of confidential for an organization. There certainly exist certain types of data such as Personally Identifiable Information, e.g., names, credit cards, social security numbers, that should be confidential regardless of the organization. The definition becomes more difficult to articulate, however, when we consider trade secrets and internal communications, which may be unstructured. Broadly, we define secret as information generated within the organization that is either not generally known, e.g., facts that can be found in an encyclopedia or industry magazines, or contained in public materials from the company. A DLP system will include some functionality to identify sensitive information in one or more of the aforementioned data types.

A DLP system performs three broad steps to prevent enterprise data loss. First, the system discovers the three types of enterprise data by scanning storage devices, intercepting network traffic in real time, and monitoring user actions on end point devices. Second, the system identifies confidential enterprise data from the data discovered in the first step. Third, the system enforces enterprise policies on confidential data. For example, the system may encrypt confidential data-at-rest to prevent unauthorized use; the system may block confidential data-in-motion from leaving the enterprise and may prevent confidential data from being copied to a USB device.

A DLP system faces two operational challenges: performance and accuracy. In an enterprise setting, the system should scan terabytes of data-at-rest, monitor hundreds of megabytes of real time network traffic, and monitor user actions on thousands of end point devices. The system should identify confidential data accurately in a scalable manner without producing many false positives or false negatives.

Current DLP products identify confidential data in three ways: regular expressions, keywords, and hashing. Regular expressions are used primarily to rec-

ognize data by type, e.g., social security numbers, telephone numbers, addresses, and other data that has a significant amount of structure. Keyword matching is appropriate when a small number of known keywords can identify private data. For example, medical or financial records may meet this criteria. For less structured data, DLP products use hash fingerprinting. The DLP system takes as input a set of private documents and computes a database of hashes of substrings of those documents. The system considers a new document private if it contains a substring with a matching hash in the database. Regular expressions are good for detecting well-structured data, but keyword lists can be difficult to maintain and fingerprint-based methods can miss confidential information if it is reformatted or rephrased for different contexts such as email or social networks.

It is also unlikely that more sophisticated access controls and additional user annotation will necessarily improve DLP products. First, it is likely that most sensitive materials contain a fair amount of public knowledge. Former analysts of the C.I.A. have noted that only 5% of intelligence was captured through covert actions, meaning that 95% of information in these reports is derived from public sources[24]. Therefore, assigning the privacy level to text copied and pasted from such a document is not guaranteed be the correct action. Relying on the users themselves to better identify and police sensitive materials poses several complications. Users may find encoding sensitive material to not be trivial. Even if the user has the ability to sufficiently define what is confidential in this system, it is possible for the user to forget or make a mistake. Lastly, it may not be feasible to expect that all users annotate their content consistently.

In this paper, we propose automatic document classification techniques to identify confidential data in a scalable and accurate manner. In our approach, the enterprise IT administrator provides a labeled training set of secret and non-secret documents to the DLP system instead of keywords and regular expression. We *learn* a classifier from the training set; the classifier can accurately label both structured and unstructured content as confidential and non-confidential. The DLP system will use the classifier to identify confidential data stored on the enterprise devices or sent through the network.

Our approach builds on a well-studied machine learning technique, Support Vector Machines (SVMs), that scales well to large data sets [30]. The classifier can meet an enterprise's needs ranging from a small collection of a user's sensitive material to a large enterprise-wide corpus of documents. We assume that the DLP system cannot access meta-data associated with documents, e.g., author, location, time of creation, and type. We also assume that administrators will only provide the document classifier with examples of confidential and non-confidential materials. Employees and managers, therefore, can provide confidential documents directly to the classifier, alleviating the burden of collecting a training set on IT administrators and minimizing their exposure to confidential information.

The major drawback of confidential data identification schemes used in DLP systems, including ours, is the inability of these systems to classify data they do not "understand." Encrypted data and multimedia content are examples of such

data. Loss of confidential data via encryption is relatively rare in practice, only 1 out of more than 200 data breaches use encryption [54]. Hence we leave the challenges of identifying confidential data in encrypted content and multimedia content as future work.

3 Text classifiers for DLP

This section will discuss present our approach for building text classifiers for Data Loss Prevention. We will begin by discussing the types of data a text classifier will encounter with respect to prominence and privacy. We will then describe our baseline approach for performance comparison. We will conclude the section with our approach to building text classsifiers for DLP.

Enterprise networks and computers handle three types of data: public enterprise data, private enterprise data, and non-enterprise data. Public enterprise data (*public*) includes public web pages, emails to customers and other external entities, public relations blog posts, etc. Private enterprise data (*secret*) may include internal policy manuals, legal agreements, financial records, private customer data, source code or other trade secrets. Non-enterprise data (*NE*) is everything else, and so cannot be described succinctly, but is likely to include personal emails, Facebook pages, news articles, and web pages from other organizations, some of which may be topically related to the business of the enterprise. We consider private documents to be confidential and require protection whereas *NE* and *public* documents do not. From this high-level description, we can draw several conclusions:

- Enterprise public and private documents are likely to be relatively similar since they discuss different aspects of the same underlying topics.
- Many non-enterprise documents will share almost no features with enterprise documents.
- Some non-enterprise documents may be quite similar to enterprise public documents. For example, non-enterprise documents may include news articles about the enterprise or web pages from related organizations.

A DLP text classifier is thus faced with two contradictory requirements: it must be finely tuned to enterprise documents so that it can make the subtle distinction between public and private documents that discuss the same topic, but it must not overfit the data so that it can correctly mark non-enterprise documents as public. As explained below, our solution uses a two-step classifier to solve this problem. The first step eliminates most non-enterprise documents that have little in common with enterprise documents, and the second step uses a classifier focused on documents related to the enterprise to make the finer distinction between enterprise public and private documents.

3.1 Baseline approach

We are not aware of any previously published results on text classification for DLP. We also could not test our solution against existing DLP solutions because

we could not verify if the software adhered to the constraints our classifier abides to (e.g. no meta-data is associated with documents). We first developed a baseline classifier to provide a basis for comparison and to garner insight into the structure of the DLP text classification problem.

We performed a brute search evaluating multiple machine learning algorithms and feature spaces known for their text classification performance for our baseline classifier, including SVMs [28], Naive Bayesian classifiers [35], and Rocchio classifiers [35] from the the WEKA toolkit [20] to determine the best classifier across all the datasets. We found that a support vector machine with a linear kernel performed the best on our test corpora (described in Section 4). The best performing feature space across all corpora is unigrams, i.e. single words, with binary weights. We eliminated stop words, common words such as "is" and "the", and limited the total number of features to 20,000. If a corpus contained more than 20,000 unique non-stop words, we choose the 20,000 most frequently-occurring non-stop words as our features. We use this configuration as our baseline classifier for all experiments reported in Section 5.

An SVM trained on enterprise documents achieves reasonable performance on enterprise documents, but has an unacceptably high false positive rate on non-enterprise (NE) documents. The poor performance can be explained by identifying weaknesses in the training approach. First, for two of our corpora, the classifier was biased towards the secret class, e.g., its initial expectation was most documents to be secret. And since many NE documents share very few features in common with secret documents, the classifier mislabeled these instances because it had too little information to contradict its a priori expectation. The second issue arose from overfitting of features. The public documents could not alone capture the behavior of these features for non-secret documents. It will, therefore, overweight certain features; we noticed common words like "policy" and "procedure" being instrumental in the misclassification of NE documents.

3.2 Supplement and Adjust

To remedy overfitting and overweighting common features, we *supplement* the classifier by adding training data from non-enterprise collections such as Wikipedia [16], Reuters[33], or other public corpora. As we will show in Section 5, our supplemental corpus does not need to be comprehensive. The presence of supplementary data does not train the classifier to recognize NE documents, but prevents it from overfitting the enterprise data.

We use 10,000 randomly-selected Wikipedia articles and a 1,100 document set featuring documents on finance, law and sport as our supplementary data set. We labeled the supplementary articles as *public* during training. The supplement classifier uses the same feature set as the baseline classifier and does not include features found in the supplemental data set. This prevents the classifier from using words from the supplemental data set to learn to distinguish enterprise and non-enterprise documents.

Adding supplemental training data will likely introduce a new problem: class imbalance. Supplemental instances will bias the classifier towards *public* docu-

ments because the size of this class will overwhelm the size of *secret* documents. This will result in a high false-negative rate on *secret* documents. Therefore, we need to adjust the decision boundary towards *public* instances. This will reduce the false negative rate while increasing the false positive rate. For our classifier, we measure the distance between the decision boundary and the closest, correctly classified *public* instance (either NE or *public*) and move the boundary $x\%$ of the distance towards it, for some value of x. We chose $x = 90\%$, although we show in Appendix A that our classifier is robust and performs well when $50\% \leq x \leq 90\%$.

The supplement and adjustment technique can be applied to train classifiers tailored to both *public* and *secret* documents, with the supplemental instances in both cases drawing from the same source, e.g., Wikipedia. Therefore, we denote a supplement and adjust classifier as SA_{class} where class is either *public* or *secret*. When training an SA_{secret} classifier, we combine *public* and NE documents and adjust the boundary to the closest, correctly classified *public* or NE. An SA_{public} classifier is constructed by combining *secret* and NE documents and adjust the boundary to the closest, correctly classified *secret* or NE document. We employ an SA_{secret} classifier as the first stage of our DLP text classification system.

3.3 Meta-space classification

The first-level classifier significantly reduces the number of false positives generated by NE documents, but not completely. These documents tend to contain salient features of the *secret* class, but upon further inspection, clearly unrelated topically to confidential documents. Also, the number of false positives for *public* documents increases. Therefore, we apply a second step to eliminate false positives from documents labeled *secret* by the first step.

We address these remaining false positives in three ways. First, for a target document, we will measure how similar it is to either the *secret* or *public* set of documents. Second, we build classifiers specifically tailored for the *public* class. *Secret* and *public* documents will likely overlap in content since they are topically related and may even discuss the same entities employing similar language. Therefore, our system will attempt to learn what combination of features make these documents *public* rather than *secret*. We can use the output of this classifier in conjunction with the first step to better gauge if a document should be labeled *secret* or not. Lastly, we classify the target document based on the output of the similarity measures and classifiers (hence why we refer to this classifier as a "meta-space" classifier). We use three classes (*public*, NE, *secret*) instead of two classes (*secret*,¬*secret*) for this step. Three classes assist the classification of *secret* documents because NE false positives exhibit different behaviors than *public* false positives for these features, making classification much more difficult if we group NE and *public* together.

To address the problem of topically unrelated documents being labeled as *secret*, we created two attributes, $xtra.info_{secret}$ and $xtra.info_{public}$, that measure the percentage of words in a document that do not appear in any document from the *secret* and *public* training corpora, respectively. These features are intended

to measure the overall dissimilarity of a document, d, to documents in the *public* and *secret* corpora. For example, if d has a large value for $xtra.info_{public}$, then it is very different from documents in the *public* training corpus. We can improve the $xtra.info$ features by ignoring words that occur commonly in English and hence convey little contextual information. We compute for each word w an estimate, df_w of how often w occurs in "general" English documents. We can then ignore all words that have a high df_w value. We used 400,000 randomly-selected Wikipedia articles to estimate df_w for all words across all our training sets. If a word in our training set never occurred in Wikipedia, we assigned it a frequency of $1/400,000$. We then computed

$$xtra.info_c(d) = \frac{\left| d_{df} \setminus \bigcup_{d' \in c} d' \right|}{|d_{df}|}$$

where $d_{df} = \{w \in d | df_w \leq df\}$. In our experiments, we used $df = 0.5\%$.

The $xtra.info_{secret}$ attribute aides the classifier by giving some context information about the document being classified. If the test document is truly *secret*, than we expect it to be similar to existing *secret* documents with respect to non-trivial language (enforced by the df threshold). Table 4 shows that for NE examples from the Wikipedia Test corpus, the $xtra.info_{secret}$ is quite high and enables a second classifier to easily separate these documents from true *secret* documents.

To better differentiate between *public* and *secret* documents, we train a SA_{public} classifier. By combining *secret* and NE documents, the classifier will better recognize which features correlate with *public* documents. On it's own, the output of the classifier will not necessarily exceed the performance of the SA_{secret} classifier. But when combined with the output of SA_{secret}, $xtra.info_{public}$ and $xtra.info_{secret}$, the classifier better discriminates between *public* and *secret* enterprise documents.

The usage of this meta-space classification is improved by using three classes instead two (i.e. *secret* or $\neg secret$). Combining *public* and NE is not optimal because we expect much different behavior for each of the attributes. NE documents will most likely have higher $xtra.info_{private}$ and $xtra.info_{public}$ scores than *public* documents and be classified $\neg public$ by SA_{public}. This will negatively affect classification for these attributes because the separability of these values is diminished by grouping them together. Our SVM uses Hastie et al [21] pairwise coupling algorithm for multiclass classification.

In summary, our meta-space classifier is trained four features: the outputs of SA_{public} and SA_{secret} classifiers, $xtra.info_{public}$ and $xtra.info_{secret}$. We train the classifier on the NE, *public*, and *secret* documents that were misclassified by SA_{public}. NE and *public* documents are not combined together as in the $SA_{private}$ classifier, but rather, assigned to one of three classes (NE, *public* and *secret*) based on its prominence. To classify a new document, d, we first compute $SA_{secret}(d)$. If this classifier indicates that d is not *secret*, we mark d as *public*. Otherwise, we compute $SA_{public}(d)$ and $xtra.info_{public}$ and $xtra.info_{secret}$ for d and apply the meta-space classifier to obtain a final decision.

4 DLP corpora

We have created five corpora for training and evaluating DLP classification algorithms. To our knowledge, these are the first publicly-available corpora for evaluating DLP systems. Constructing DLP corpora is challenging because they should contain private information from some enterprise, but private information is, by definition, difficult to obtain.

Three of our corpora – DynCorp, TM, and Mormon – contain private documents leaked from these organizations to WikiLeaks and public documents taken from the organizations' public web sites. We collected 23 documents from DynCorp, a military contractor, from their field manual for operatives. We obtained 174 web pages from their website [15]. WikiLeaks hosts 102 documents from Transcendental Meditation, a religious organization, that include workshop instructions written by high-ranking members of the organization. We obtained 120 public materials from various TM affiliated websites [38]. The Mormon corpus includes a Mormon handbook that is not to be distributed outside of its members. We split the handbook into 1000 character-long pieces and added two other smaller supplemental organizational documents from the church available through WikiLeaks. We split the document into smaller chunks since the handbook is the main document we could obtain from this organization, but it is also one of the most sensitive documents the organization possesses. We took an arbitrary split of 1000 characters since it should provide enough textual information for classification. We gathered 277 webpages from the Church of Jesus Christ of Latter Day Saints website [42]. Note that our inclusion of texts from religious organizations is not intended to denigrate these faiths, but because they are documents that these organizations tried to keep secret.

Our Enron corpus contains emails released during the Federal Energy Regulatory Commission labeled by Hearst et al. [23]. Our data set only includes "business-related" emails. Since Enron is now defunct, we used the Internet Archive to obtain documents from its public website [3]. We were able to obtain 581 web pages.

The Google private document dataset consists of 1119 posts by Google employees to software-development blogs. Google collaborates with many open-source software projects, so much of its software development discussions take place in public. If these same projects were conducted as closed source development, then these blog posts would be private, internal documents, so we treat them as such in our dataset. 1481 public documents were taken from PR-related blogs.

Finally, we include several corpora that are intended to represent non-enterprise documents. We sampled 10K randomly selected Wikipedia articles and denote it as the Wikipedia Test corpus. We also test the robustness of our classifier on the Brown [2] (500 samples) and Reuters [33] corpora (10788).

	DynCorp		TM		Enron		Mormon		Google	
Classifier	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
Baseline	0.0%	0.0%	2.5%	0.98%	0.87%	0.0%	0.72%	1.4%	1.8%	1.9%
Supplement	0.0%	8.0%	0.0%	11.0%	0.0%	5.0%	0.0%	0.3%	0.0%	3.7%
Supplement and Adjust	2.0%	0.0%	28.3%	0.0%	4.1%	1.2%	4.6%	0.0%	15.9%	0.3%
Two-step	0.0%	0.0%	0.0%	0.98%	0.87%	3.0%	0.36%	1.4%	1.0%	2.1%

Table 1. The false positive (FP) and false negative (FN) rates on the enterprise corpora for each of our classification strategies. 11,100 instances and an adjustment of 90% are used.

Non-enterprise False Positive Rate					
Classifier	DynCorp	Enron	Mormon	Google	TM
Baseline	4.7%	87.2%	0.16%	7.9%	25.1%
Supplement	0.0%	0.01%	0.06%	0.0%	0.0%
Supplement and Adjust	0.26%	2.5%	0.1%	2.8%	0.93%
Two-step	0.0%	0.05%	0.0%	0.06%	0.01%

Table 2. The false positive rates on our Wikipedia false positive corpus for each of the classification strategies.

5 Evaluation

A successful DLP classifier must meet several evaluation criteria. It must have a low false negative rate (i.e. misclassifying *secret* documents) and low false positive rate for any non-*secret* document. It should also achieve a low false discovery rate. Furthermore, we need to show that our classifier is robust with respect to its training parameters, in particular: the choice of the supplemental corpus, the size of the supplemental corpus, and the degree of adjustment used in the supplement and adjust classifier.

We present the results of our training strategy against a baseline classifier. For all our classifiers, we tokenize all our datasets and use unigrams for features. For a baseline classifier, we only train the classifier on enterprise documents using the binary weighting scheme. For the results presented in Tables 1 and 2, we supplement the classifiers with 10000 Wikipedia articles and 1100 topical articles and adjust the classifier to move the decision boundary 90% of the distance between the decision boundary and the closest correctly labeled *public* instance. We use a document frequency of 0.5% to compute $xtra.info_{secret}$ and $xtra.info_{public}$. We compute the false negative and false positive rates by performing a 10-fold cross validation on each of the corpora, and then determine the false positive rate for NE documents by training the classifier on the entire enterprise dataset and then classifying our Wikipedia false positive corpus.

The results of our classification tests show that our training strategy maintains low false negative and false positive rates on enterprise documents while dramatically improving the false positive rate on NE documents. The baseline

Dataset	Baseline FDR	Our classifier FDR
DynCorp	4.49%	**0.00%**
Enron	47.05%	**0.92%**
Google	8.99%	**1.06%**
Mormon	0.88%	**0.36%**
TM	22.06%	**0.01%**

Table 3. The False Discovery Rate of the baseline approach far exceeds our classifier, implying that the baseline approach would fare poorly in real world networks whereas ours would not raise much fewer alarms.
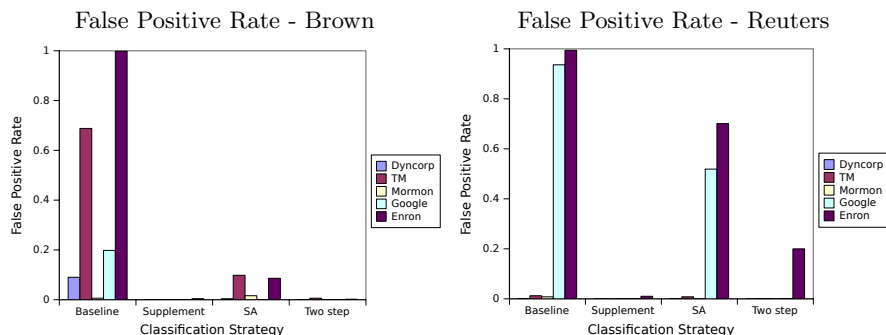
Fig. 1. The false positive rates for each classification strategy. The two-step classifier is able to maintain a low false positive rate across all the different corpora for each *non-enterprise* corpora.

approach would be unusable in practice because of its high false positive rate on NE documents.

In our results shown in Table 3, we assume the following traffic composition in a typical enterprise network: 25% enterprise secret documents, 25% enterprise public documents, and 50% non-enterprise documents. We believe that our approach will not engender "alarm fatigue", whereas the baseline approach is likely to overwhelm operators with false alarms.

The supplement and adjust classifier achieves a low false positive rate on NE documents for several reasons. The supplement and adjustment classifier did not rely on finding features that were strongly indicative of the *public* class. This is a crucial benefit because the NE document set's size is so large that it would be impossible to create a set of features that were *strongly indicative* of all possible *public* documents. In addition to relying less on features that were indicative of *public* documents, the supplement and adjustment classifier moves the expectation further towards the *public* class, which is in line with our expectation of the problem outlined in the problem description. And by performing an adjustment to the decision boundary, the classifier reduces the false negative rate without increasing the false positive rate, when combined with the second level classifier.

5.1 Effective training parameters

Figure 1 demonstrates that our classifier is robust with respect to the choice of the supplemental corpus. Our supplemental corpus consisted solely of Wikipedia documents but, as Figure 1 shows, the resulting two-step classifier has a low false positive rate on NE documents drawn from drastically different corpora, such as the Brown or Reuters news corpora. Thus, we can build a standard non-enterprise corpus that is used by all enterprises to train their DLP systems. The corpus will not need to be customized for each enterprise or for each new form of Internet traffic.

As expected, a larger supplemental corpus decreases the false positive rate but increases the false negative rate as the classifier becomes more biased towards *public* documents (see Appendix A for details). Note that Google is a clear outlier in this evaluation. We suspect that this may be because the Google corpus is the only artificial corpus in our data set. Recall that all the Google documents, including the "private" ones, are in reality *public* documents, unlike our other corpora which contain genuine private enterprise documents. The second step of our approach remedies the errors made on *public* enterprise documents. We also conclude that the supplemental corpus does not need to be too large – about 10,000 documents suffice.

We also investigated the effect of the adjustment value on the classifier. According to the graphs in Appendix A, an adjustment value of 0.5 provides a good trade-off between increased false positives and false negatives in the supplement and adjust classifier. However, since we added a second-level classifier that can filter out many false positives, we chose an adjustment value of 0.9 in order to achieve a slightly lower false negative rate.

6 Discussion

The algorithm presented in this paper should prevent accidental leakages of information, but how will it fare against intentional leakages? According to Proof-Point [44], most data leakages are accidental. The most common intentional leakage occurs when employees download sensitive information upon termination of employment. Our method coupled with the DLP system's ability to recognize data flow from a trusted to an untrusted device should prevent these type of leakages. If the data were encrypted or re-encoded, this would exceed the capability of our classifier. These more sophisticated attacks, fortunately, only account for 1 in 200 data breaches [54].

It is instructive to highlight key differences between our solution and existing semi-supervised and class imbalance solutions. Our algorithm is a supervised learning approach: all examples are labeled. During training, the classifier will know if the enterprise document is confidential or not. Since supplemental training instances do not come from the enterprise, these instances are labeled opposite from the class we wish to train on, e.g., for the $SA_{private}$ classifier, these supplemental instances are labeled as *public*. For the purposes of our algorithm,

we focus on recognizing sensitive information that either it has either seen before or is similar to an existing confidential document. In the future, we hope to explore how the system can infer if a document is sensitive if it has zero training data to support this decision (possibly relying on metadata).

Our study demonstrates that DLP systems face an inherent class imbalance issue: nearly all documents that exist are outside the organization and are not sensitive. To train a classifier on this class is simply infeasible because of its size. Our key insight into this problem is recognizing that our classifiers needed to be trained to effectively learn what is *secret*, and not rely too heavily upon features that were correlated with non-*secret* documents. The problem of class imbalance has been studied and work in this area is discussed in Section 7. Once we recognized that class imbalance would be an issue for achieving maximal performance, we tried many of the approaches listed in the Section 7, but found that they were ineffectual on this specific problem.

Our approach is unique from other class imbalance techniques because we attempt to better determine which features correlate with sensitive information by adding additional samples that express a diverse usage of language. We cannot say how well this technique will extrapolate to other machine learning problems, but it is applicable to our specific problem of generating a classifier robust enough to perform well in the presence of many unrelated documents. To the best of our knowledge, using supplemental data (not synthetically generated) to generate negative examples has not been applied to the class imbalance for text classification.

An important design decision in this algorithm was to restrict the vector space to features included only in *secret* and *public* documents. The reasoning behind this decision is related to the class imbalance aspect of this problem. Since the number of non-*secret* documents is so large, adding additional features to the vector space would have resulted in overfitting because those features would factor prominently into classifying NE documents in the training step. The classifier may not accurately reweight features that *secret* documents share with non-*secret* documents. And since it would be impossible to provide the classifier with training representative of everything that is NE, the classifier would be more likely to generate false positives.

The $xtra.info$ attribute performs exceedingly well in maximizing separability between NE and *secret* documents, as shown in Table 4. Contextual information is quite important because we have limited our vector space to only enterprise documents, which these terms are assumed to be related to the knowledge domain of the enterprise. Using a unigram vector space, we lose contextual information that may help counteract the effect of polysemy that contributes to the misclassification of NE documents. Our $xtra.info$ attribute is effective in the second level of classification in providing contextual information to disambiguate between *secret* and NE classes and is easily computable.

The techniques of our algorithm performed well for many different different types of media and organizations. One limitation in creating our DLP corpora is that it the documents for each organization do not represent the entirety of its

Mean $xtra.info_{secret}$	Dyncorp	Enron	Google	Mormon	TM
Secret documents	0.54 (0.10)	0.83 (0.09)	0.70 (0.15)	0.49 (0.15)	0.66 (0.11)
NE documents	0.96 (0.03)	0.99 (0.02)	0.98 (0.04)	0.95 (0.08)	0.99 (0.02)

Table 4. This table presents the means for the $xtra.info_{secret}$ attribute for each of our private corpora and the document classes *secret* and NE. The significant differences between the means for these classes suggest that this attribute will aide the classifier in distinguishing NE documents from *secret*.

operations. It was not feasible to either find or build a corpus of this nature because of the risk for corporations assembling and releasing this data. We believe, however, that since our algorithm performed well in many different instances, it will perform well enterprise wide. Depending on the size and structure of the organization, multiple classifiers can be built for each of the different departments and group. Text clustering can also assist in building cogent collections of documents to train and build classifiers. And since the classification techniques we use are not computationally expensive, the penalty for evaluating multiple classifiers is not prohibitively greater.

The system described in this paper will most likely be part of a larger enforcement framework that will defend the network from intrusions and malware. Administrators will need to provide instances of *secret* and *public* documents because the training of our system is supervised. This collection of samples, however, should not be difficult to obtain because it can be automated and does not require further annotations. Employees can designate folders on storage devices that contain either *secret* or *public* documents or manually submit examples through a collection facility, e.g, email or web-based system. Depending on the enterprise policy enforcement guidelines, messages that the classifier suspects to be *secret* may prompt the sender to reconsider, queue the message for an administrator to review, or simply block the transaction. The toolkit we implemented will be made available from the author's website.

7 Related work

Automated document classification is a well studied research area. Research in the document classification field dates back to 1960s [36, 6]. The use of machine learning in text classification, however, became popular in the last two decades. Sebastiani provides an excellent overview of the area: he describes various text categorization algorithms, approaches to evaluate the algorithms, and various application of automated text categorization [48]. The proliferation of digital documents and the explosion of the web has given rise to many applications of document classification, e.g., automatic document indexing for information retrieval systems [18], classification of news stories [22], email filtering to classify emails into categories [12], spam filtering to identify spam from legitimate email messages [1], automatic categorization of web pages [4, 8], and product review

classification [51]. The research community has explored many different machine learning approaches for text categorization, e.g., Bayesian classifiers [1, 34, 32], decision trees [7], k-nearest neighbors [14], neural networks [41], regression models [17], and support vector machines [29]. Researchers have also experimented with the idea of combining multiple classifiers to increase efficacy, most notable being Schapire et al.'s *boosting* approach [47].

We utilize Support Vector Machines, a powerful margin-based classification and regression technique introduced by Cortes and Vapnik, in our classifier [13]. Joachims applied SVMs to the text classification task [28] and identified properties of text categorization that makes SVMs suitable for the task. For example, text categorization has to deal with large numbers of features, as words present in a document are considered the document's features. Feature selection is a traditional approach to select a few *relevant* features from many. In the case of text, however, most features are relevant. Hence a good text classifier should be able to handle many features. SVMs can handle large numbers of features due to overfitting protection. Also, SVMs are good *linear* classifiers and many text categorization tasks are linearly separable.

Text classification for DLP presents difficulties that standard classifiers cannot solve because of the lack of a proper training set. It is difficult to supply the classifier with an adequate representation of what should be *public* (i.e., not *secret*). Therefore, this paper addresses the precise problem of an unrepresentative dataset for text classification with the techniques of supplement and adjust, $xtra.info$, and utilizing a two-step classifier. Other research has focused on related topics.

Accurate text classification in the case of limited training examples is a challenging task. Joachims used a transductive approach to handle the problem [31]; his approach focuses on improving the classification accuracy of SVMs for a given test set. Blum and Mitchell introduced a co-training approach to categorize web pages [5]. They improved classification accuracy by adding a larger number of unlabeled examples to a smaller set of labeled examples. Toutanova et al. demonstrated the use of hierarchical mixture models in the presence of many text categories [52].

Researchers have also investigated mitigating the effect of class imbalance on classification performance [10]. Both oversampling and undersampling classes in the training instances has been widely investigated. The sampling can be random or directed. Synthetic generation of examples for underrepresented classes has also been explored and combined with under and over sampling [9]. One class learning classifiers have been proposed to improve classification for target classes where examples are relatively scarce compared to other classes [27]. An instance is compared with training examples in terms of similarity to determine whether the instance is a member of the target class. Lastly, feature selection techniques can improve classification of underrepresented classes because high dimensional data may overfit or be biased towards the majority classes [40].

Several projects have used Wikipedia to enhance text classification, particularly where context is unavailable due to the brevity of the text to be classified

[19, 43, 25, 39]. Gabrilovich et al. [19] first proposed transforming a document into its representation in Wikipedia topic space. Others have modified this basic idea by including topic hyponomy and synonymy [25] or performing LSA on this topic space [39]. Others have investigated using Wikipedia to determine relatedness between texts, particularly short texts [49]. To our knowledge, no one has investigated using Wikipedia explicitly to augment their training corpus.

8   Conclusion and Future Work

This paper presents a simple, efficient, and effective way to train classifiers and perform classification for Data Loss Prevention. In doing so, it presents the first corpora for the DLP task. Our results indicate a naive approach to training a classifier, solely on documents from the enterprise, will lead to a high false positive rate on unrelated documents, indicating poor real world performance. The paper presents a novel technique, supplement and adjust, which reduced the false positive rate for documents unrelated to the core business function.

We plan to further study the efficacy of our text classification approach by deploying it on existing private, enterprise and governmental networks. We will also look to expand our approach to include encrypted and multimedia content. In this work, we only consider the content of a document to render a decision. We would like to investigate what meta data associated with the content could be used to improve classification.

Lastly, not all secret documents in the world are written in English. We will hope to expand our private corpus in the future to include non-English sources. Our intuition is that many language processing techniques developed to handle language specific obstacles should be applied to the processing of these documents. We will also have to adjust our supplemental corpus accordingly to provide realistic behavior for NE feature behavior.

References

1. Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, and Constantine D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167, New York, NY, USA, 2000. ACM.
2. Internet Archive. Brown corpus. `http://www.archive.org/details/BrownCorpus`.
3. Internet Archive. Wayback machine. `http://www.archive.org/web/web.php`.
4. Giuseppe Attardi, Antonio Gull, and Fabrizio Sebastiani. Automatic web page categorization by link and context analysis. In *Proceedings of THAI-99, 1st European Symposium on Telematics, Hypermedia and Artificial Intelligence*, 1999.
5. Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, New York, NY, USA, 1998. ACM.

6. Harold Borko and Myrna Bernick. Automatic document classification. *J. ACM*, 10(2):151–162, 1963.

7. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

8. Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *The VLDB Journal*, 7(3):163–178, 1998.

9. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Inteligence Research*, 16:321–357, June 2002.

10. Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kolcz. Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explorer Newsletter*, 6(1), 2004.

11. Privacy Rights Clearinghouse. Chronology of data breaches: Security breaches 2005–present. `http://www.privacyrights.org/data-breach`, August 2010.

12. William W. Cohen. Learning rules that classify e-mail. In *In Papers from the AAAI Spring Symposium on Machine Learning in Information Access*, pages 18–25. AAAI Press, 1996.

13. Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

14. T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 1967.

15. Dyncorp. Dyncorp website. `http://www.dyncorp.com`.

16. Wikimedia Foundation. Wikipedia. `http://en.wikipedia.org/`.

17. David Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, 2005.

18. N. Fuhr and G. E. Knorz. Retrieval test evaluation of a rule based automatic indexing (air/phys). In *Proc. of the third joint BCS and ACM symposium on Research and development in information retrieval*, pages 391–408, New York, NY, USA, 1984. Cambridge University Press.

19. Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1606–1611, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

20. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

21. Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, pages 507–513, Cambridge, MA, USA, 1998. MIT Press.

22. Philip J. Hayes and Steven P. Weinstein. Construe/tis: A system for content-based indexing of a database of news stories. In *IAAI '90: Proceedings of the The Second Conference on Innovative Applications of Artificial Intelligence*, pages 49–64. AAAI Press, 1991.

23. Marti Hearst. Teaching applied natural language processing: triumphs and tribulations. In *TeachNLP '05: Proceedings of the Second ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 1–8, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

24. Frederick Hitz. *Why Spy?: Espionage in an Age of Uncertainty*. Thomas Dunne Books, 2008.

25. Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, and Zheng Chen. Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 179–186, New York, NY, USA, 2008. ACM.

26. Poneman Institute. Fourth annual us cost of data breach study. `http://www.ponemon.org/local/upload/fckjail/generalcontent/18/file/2008-2009USCostofDataBreachReportFinal.pdf`, January 2009.

27. Nathalie Japkowicz. Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning*, 42:97–122, January 2001.

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

29. T. Joachims. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer, 2002.

30. Thorsten Joachims. *Making large-scale support vector machine learning practical*, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

31. Thorsten Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*, pages 200–209, Bled, Slowenien, 1999.

32. D Koller, U Lerner, and D Angelov. A general algorithm for approximate inference and its application to hybrid bayes nets. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence*, 1999.

33. David Lewis. Reuters 21578. `http://www.daviddlewis.com/resources/testcollections/reuters21578/`.

34. Xiaoli Li and Bing Liu. Learning to classify texts using positive and unlabeled data. In *IJCAI'03: Proceedings of the 18th international joint conference on Artificial intelligence*, pages 587–592, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

35. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

36. M. E. Maron. Automatic indexing: An experimental inquiry. *J. ACM*, 8(3):404–417, 1961.

37. McAfee. Data loss prevention. `http://www.mcafee.com/us/enterprise/products/data_loss_prevention/`.

38. Transcendental Meditation. Transcendental meditation websites. `http://www.alltm.org` and `http://www.tmscotland.org`.

39. Zsolt Minier, Zalan Bodo, and Lehel Csato. Wikipedia-based kernels for text categorization. In *Proceedings of the Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 157–164, Washington, DC, USA, 2007. IEEE Computer Society.

40. Dunja Mladenic and Marko Grobelnik. Feature selection for unbalanced class distribution and naive bayes. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 258–267, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

41. Kenney Ng. A comparative study of the practical characteristics of neural network and conventional pattern classifiers. Technical report, 1990.

42. Church of Jesus Christ of Latter Day Saints. Church of jesus christ of latter day saints website. `http://lds.org`.

43. Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 91–100, New York, NY, USA, 2008. ACM.

44. Proofpoint. Outbound email security and data loss prevention. `http://www.proofpoint.com/id/outbound/index.php`.

45. proofpoint. Unified email security, email archiving, data loss prevention and encryption. `http://www.proofpoint.com/products/`.

46. RSA. Data Loss Prevention. `http://www.rsa.com/node.aspx?id=1130`.

47. Robert E. Schapire. Theoretical views of boosting and applications. In *ALT '99: Proceedings of the 10th International Conference on Algorithmic Learning Theory*, pages 13–25, London, UK, 1999. Springer-Verlag.

48. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

49. Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1419–1424. AAAI Press, 2006.

50. Symantec. Data Loss Prevention Products & Services. `http://www.symantec.com/business/theme.jsp?themeid=vontu`.

51. Tun Thura Thet, Jin-Cheon Na, and Christopher S. G. Khoo. Filtering product reviews from web search results. In *DocEng '07: Proceedings of the 2007 ACM symposium on Document engineering*, pages 196–198, New York, NY, USA, 2007. ACM.

52. Kristina Toutanova, Francine Chen, Kris Popat, and Thomas Hofmann. Text classification in a hierarchical mixture model for small training sets. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 105–113, New York, NY, USA, 2001. ACM.

53. Trend Micro. Trend Micro Data Loss Prevention. `http://us.trendmicro.com/us/products/enterprise/data-loss-prevention/`.

54. Trustwave. Global security report 2010. `https://www.trustwave.com/whitePapers.php`, February 2010.

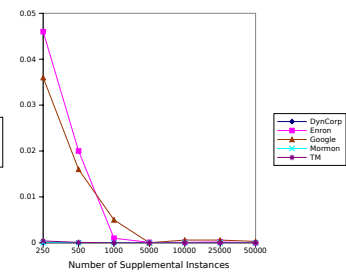# A   Effects of Supplement and Adjustment
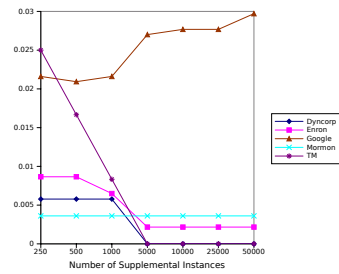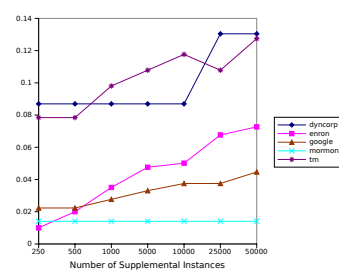
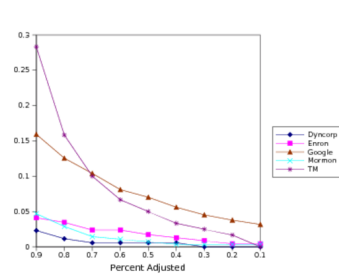## Public False Positive Rate — Secret False Negative Rate — NE False Positive Rate
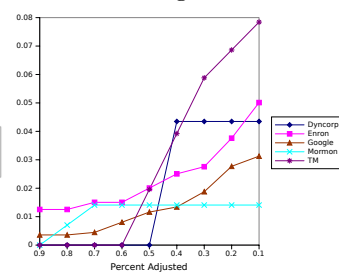
The effect on the false negative and false positive rates for our corpora when supplementing the training instances with Wikipedia examples. For the Mormon corpus, the effect of adding any supplemental instances seems to affect the classification of the same documents.
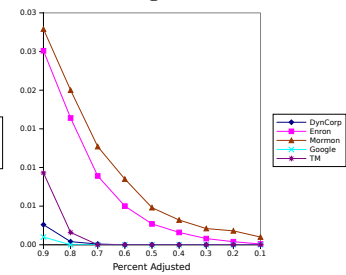

Public False Positive Rate — Secret False Negative Rate — NE False Negative Rate

The false positive and false negative rates on enterprise documents after applying the supplement and adjust classifier.